AD-A070 223 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2

COMPUTER NETWORKS.(U)

MAR 79 R B KURTH

UNCLASSIFIED NL

1 OF 2

AD

A070223

9

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER NETWORKS

by

Ronald Brent Kurth

March 1979

Thesis Advisor: N. F. Schneidewind

Approved for public release; distribution unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Computer Networks,		5. TYPE OF REPORT & PERIOD COVERED (9) Master's Thesis, March 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Ronald Brent/Kurth		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE (11) Mar[redacted]79
		13. NUMBER OF PAGES (12) 133 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computers
Computer system
Computer networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer networks are being used in many varied applications. Chapter I of this thesis ~~will look at~~ examines network topologies, components, and design goals. The remaining three chapters are dedicated to a case study of the Naval Postgraduate School's (NPS) computer system. Computer system performance measuring tools for terminal oriented systems are discussed and one is

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 450

20. (continued)

selected for the study. The NPS system is analyzed, shortcomings identified and possible solutions suggested. Finally, two network designs are proposed for a terminal oriented system that is suitable for satisfying the needs of the NPS.

Accession For	
NTIS GRA&I	☑
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

Approved for public release; distribution unlimited

Computer Networks

by

Ronald Brent Kurth
Lieutenant, United States Navy
B.S., Chico State College, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

Author Ronald B Kurth

Approved by Norman F. Schneidewind
Thesis Advisor

F Burkhead
Second Reader

Chairman, Department of Computer Science

Dean of Information and Policy Sciences

ABSTRACT

Computer networks are being used in many varied applications. Chapter I of this thesis will look at network topologies, components, and design goals. The remaining three chapters are dedicated to a case study of the Naval Postgraduate School (NPS) computer system. Computer system performance measuring tools for terminal oriented systems are discussed and one is selected for the study. The NPS system is analyzed, shortcomings identified and possible solutions suggested. Finally, two network designs are proposed for a terminal oriented system that is suitable for satisfying the needs of the NPS.

TABLE OF CONTENTS

I. COMPUTER NETWORKS-- 7

A. TECHNIQUES OF INTERCONNECTION--------------------- 7

B. SYSTEM TOPOLOGY------------------------------------ 9

C. BUILDING BLOCKS OF A NETWORK---------------------- 14

D. NETWORK DESIGN GOALS------------------------------- 25

II. COMPUTER PERFORMANCE MEASUREMENTS FOR TERMINAL ORIENTED SYSTEMS--------------------------------------- 41

A. USER BEHAVIOR AND RESPONSE TIME------------------- 41

B. SYSTEM MEASUREMENT TOOLS--------------------------- 44

C. GETTING STARTED IN PERFORMANCE MONITORING---------- 51

III. NPS CASE STUDY-------------------------------------- 53

A. THE PRESENT ENVIRONMENT---------------------------- 53

B. USER BEHAVIOR ON CP/CMS---------------------------- 62

C. ANALYSIS PROCEDURE--------------------------------- 75

D. DATA COLLECTION------------------------------------ 84

IV. NETWORK DESIGNS FOR NPS ENVIRONMENT------------------106

A. INTRODUCTION---------------------------------------106

B. DESIGN OBJECTIVES----------------------------------106

C. DESIGN 1---114

D. DESIGN 2---120

APPENDIX A MAJOR SOFTWARE USED ON IBM 360/67 at NPS--------------------------------------128

APPENDIX B ADDITIONAL SOFTWARE RESOURCES------------------129

APPENDIX C ACCOUNTING AND MEASUREMENT---------------------130

LIST OF REFERENCES---131

INITIAL DISTRIBUTION LIST----------------------------------133

INTRODUCTION

The Naval Postgraduate School (NPS) has operated with basically the same computer system for the last 12 years. In order for the military to keep pace with society in the fields of education, research, and changing technology, NPS must offer its students, researchers, and faculty the most modern computing machinery available. It must be organized in the most efficient manner so that the users may be able to obtain maximum benefits from the facility.

This thesis will assume that this organization can only be provided by a network architecture.

In order to attack this task successfully, networks will be looked at in some detail including design philosophy in Chapter I. The next step will be to ascertain what type of user needs must be fulfilled in order to have a successful system.

To do this, first performance measuring tools will be discussed in Chapter II along with human engineering impacts on performance so that the NPS system may be analyzed. Secondly, the system will be analyzed in Chapter III presenting background qualitative and quantitative information.

Finally the design goals will be specified and two designs presented in the fourth chapter using information gathered from the other chapters.

I. COMPUTER NETWORKS

The greatest motivation for computer networks is, regardless of the main purpose of the computer system, sharing of expensive computers. The resources which may be shared are data bases, processors and software. By linking together special and general purpose computing machinery the high cost can be shared by all the users of the network thus making the system cost effective. Also, a wide range of computing machinery becomes available to users who would not ordinarily be able to financially justify purchasing a special purpose computer for for their application.

There are many definitions for computer networks. The following is one definition:

> "A computer network, also called a computer-communication network is, in the broad sense, any system composed of one or more computers and terminals, communication transmission facilities, and specialized or general purpose hardware to facilitate the flow of data between terminals and processors. Its parts consist of host processors, communication devices, transmission lines and a set of rules (communication protocols), implemented in either hardware or software, to insure the orderly flow of traffic in the network." (Ref. 16)

A. TECHNIQUES OF INTERCONNECTION

Interconnecting computers is not an easy thing to do. Networks may have several different special purpose computing machines as well as different peripheral devices. There are also software incompatibilities to consider. Nevertheless, there are many successful networks in use.

There are basically three techniques for interconnecting computers, each with its own distinguishing characteristics. They are circuit switching, message switching, and packet switching. Each allocates computer resources in a different fashion in support of communication.

<u>Circuit switching</u>

In circuit switching, a complete circuit or route is established before communication begins. This type of switching was developed for telephone systems. In this mode, the user dials a sequence of digits to obtain access to a particular computer system. He then waits until an acceptable path can be provided. In circuit switching, a message cannot be queued for later transmission. The ability to queue process and transmit a message based on information in the message is important for many computer networks. Therefore, circuit switching is seldom employed.

<u>Message switching</u>

In message switching, an individual message (a logical unit of information from the users point of view) is separately switched at each node along its path from source to destination, on the basis of information it carries with it. Each message, divided into blocks of data, is received at a node in its entirety before the next message can be received. All blocks of the message must be transmitted in their correct sequence; the receiving node rebuilds the message and acknowledges its receipt to the sending node. Only the first block contains further routing information. If the message is not accepted by the receiving node, the sending node will continue to transmit

until accountability can be verified by the receiving node or link failure detected. Direct access storage devices are utilized at nodes to buffer messages permitting unrestricted message lengths and preventing nodes from overloading and thus becoming a bottleneck in the system. For this reason message switching is sometimes referred to as store and forward communication.

Packet switching

In packet switching each message is divided into blocks or packets as with message switching but unlike message switching each packet includes control information to direct the packet across the network independently of and in parallel with, other packets. Through a complex evaluation of individual packets and switch load, circuit loading can be ascertained and the packet routed along a minimally loaded route. In packet switching, as in message switching, each packet will be retransmitted until received correctly and the final destination node will send an acknowledgement message to the origin node when the last packet is received.

The ability to immediately transmit a packet without waiting for the complete message and the ability to adjust dynamically to varying load factors minimizes resource requirements at each node. Nodes which become saturated, however, usually reject traffic until their load is normalized.

B. SYSTEM TOPOLOGY

There are three basic organizations of networks, centralized, decentralized, and distributed, which is really a special case

of the second type.

Centralized network

The centralized network is the simplest arrangement that includes switching. In Fig. (1a) the centralized network or essentially a star configuration is shown with links radiating from a single node. Each link is dedicated to a terminal or a concentrator multiplexing a cluster of terminals. (Fig. 1b)

The realiability of the central network depends heavily on the central switch. If it fails, the network stops functioning, whereas if any link or terminal fails only units local to that link fail. Any significant increase in reliability can only be obtained thru redundancy of the central switch and/or links.

Decentralized network

The distinction between centralized and decentralized networks lies in the organization of the switching function and the absence of a single controlling node. (Fig. 1c)

The added reliability of the decentralized network comes from the dispersed switching power of the system. If a switch fails and redundant paths are designed into the system, messages may be routed around the failed switch; maintaining network operation at some degraded level. Of course, such redundancy comes with the expense of additional computers (nodes) and corresponding connecting links.

Distributed network

When there are at least two disjoint paths between every pair of nodes, the network is called distributed. The ring is the simplest case of a distributed network where each node is

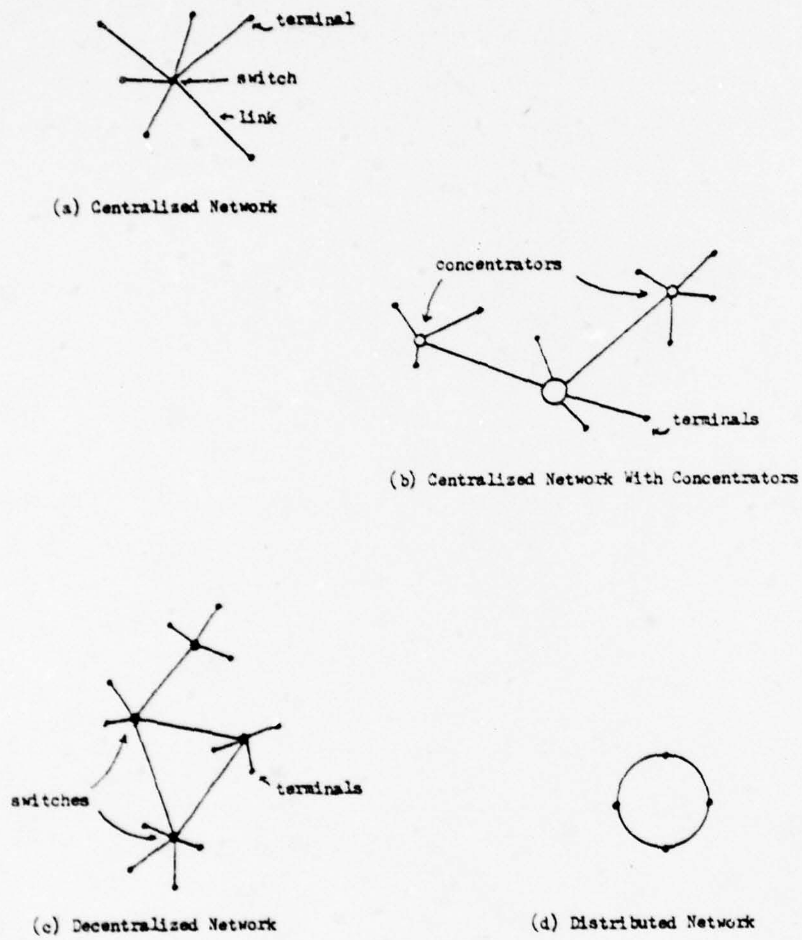


FIGURE 1 - NETWORK TOPOLOGY

connected to exactly two other nodes. (Fig. 1d). Because of the inherent reliability of distributed networks in conjunction with packet switching, this combination has the greatest potential for future networks. (Ref. 18)

Performance of these systems is characterized in terms of cost, throughput, response time, and reliability. The design of a distributed network should take into account the properties of its nodes as well as the system topology. Some of these properties are listed below, and certainly should not be limited to just distributed systems;

Node characteristics

- message handling and buffering
- error control
- flow control
- routing
- node throughput
- node reliability

Topological characteristics

- link location
- link capacity
- network response time
- network throughput
- network reliability

DISTRIBUTED PROCESSING

Along with the term distributed network, the concept of distributed processing is surfacing. With the advances in the LSI and microprocessor fields the capability of distributive

processing is here. Many vendors and authors claim wonderful benefits such as;

- high system performance, fast response, high thruput
- high availability
- high reliability
- reduced network costs
- graceful degradation (fail-soft capability)
- resource sharing
- automatic load sharing
- high adaptability to changes in work load
- ease of modular, incremental growth and configuration flexibility
- incremental replacement and/or upgrading of components (hardware and software)
- easy expansion in both capacity and function
- easy adaptation to new functions
- good response to temporary overloads

A good definition of distributed processing contains five basic components:

- A multiplicity of general-purpose resource components, including both physical and logical resources, that can be assigned to specific tasks on a dynamic basis. Homogeneity of physical resources is not essential.
- A physical distribution of these physical and logical components of the system interacting through a communication network.

- A high-level operating system, whether it exists as a distinct and identifiable block of code or only as a design philosophy, that unifies and integrates the control of the distributed components. Individual processors each have their own local operating system, and these may be unique.

- System transparency, permitting services to be requested by name only. The server does not have to be identified.

- Cooperative autonomy, characterizing the operation and interaction of both physical and logical resources.

These properties and operating characteristics are present in a number of systems to varying degrees. However, only the combination of all of the criteria uniquely defines distributed data processing systems. (Ref. 23). Users should carefully study systems which advertise "distributed processing" and determine to what degree they are distributed.

C. BUILDING BLOCKS OF A NETWORK

Most networks are made up of components which are common to every network. While not all components will be in every network at least some subset will be present.

In this section some of the more prevalent components will be briefly discussed.

TERMINALS AND TERMINAL SELECTION

The computer terminal, whether a CRT display or teletypewriter is frequently the least expensive device in an expensive

hardware system and consequently is not given much evaluation in the selection process. Yet, this single item may determine the ultimate success of the system. The terminal is the face of the computer and is for many users the only contact they will have with it. Therefore, this man-machine interface should be studied carefully.

Heavy user participation should be included in any model for terminal selection. Fig. 2 suggests a general model for selection of terminals.

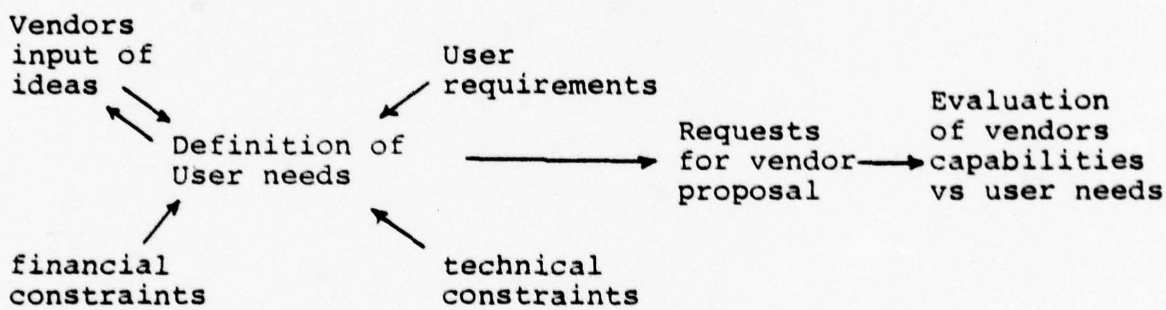


Figure 2

The major areas of general concern to most users are 1) hardware considerations, 2) special optional requirements, 3) reliability, 4) vendor marketing services, 5) vendor consultant and educational services, 6) financial considerations, 7) software and language support.

Many of the technically difficult considerations are decided by the choice of the major system. General information pertaining to terminals which are compatible to that system may be obtained from the vendor. Major hardware considerations such as scroll capability, switch selectable speed options,

size of screen, number of characters representable on the screen, data entry keyboard, and cursor control should be addressed by the users. The advantages and disadvantages of various features of terminals can be explained and demonstrated by the vendor.

The special options should also be evaluated by the users. Such things as attachable cassette tape reader-writer cartridges, slave printers for hard copy output, floppy desk storage units, and built in modems should all be explored to determine what items are desirable for current and future needs.

The question of reliability can best be approached by comparing Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) figures which are available from vendor engineering departments. These will answer the questions 1) how often does the terminal malfunction on the average, and 2) how long does it take to repair it, on the average.

Users who are evaluating terminals will also find it valuable to investigate the quality and response time of local vendor support.

The next area of concern are the consultant and educational services offered by the vendor. These may be unbundled in which case these services will have to be paid separately for by the organization. The availability and cost of these services should be carefully considered. The vendor should also be evaluated as to his experience and expertise in the area of interest. Users should obtain samplers of technical manuals in order to evaluate the vendor insofar as quality of technical support and its availability.

From the financial viewpoint, such information as 1) date of announcement of terminal, 2) date of first installation, 3) number installed to date and 4) date on which last major announcement for terminals was made by vendor should be obtained. This knowledge will help to determine obsolescence and product maturity factors. Vendor maintenance costs should also be considered in order to predict future maintenance costs.

Once the answers to these questions have been obtained, terminals can then be compared to determine which best suits the organization.

A methodology for this evaluation is suggested below:

1. Determine the meaning, relative to the organization, of each of the major areas. Evaluators should state precisely what is to be considered in terms of hardware, special optional requirements, reliability, vendor marketing services, vendor educational services, and financial considerations.
2. Assign a relative weight to each of these areas.
3. Break each of the major areas into easily quantifiable basic elements for understanding the larger more complex major area.
4. Obtain answers to all questions.
5. Evaluate the total performance of each terminal in terms of its capability to meet the organization's needs.
6. Determine the price/performance of each terminal.
7. Select the most attractive terminal.

Table 1 is a sample comparison using such a methodology.

TABLE 1

A COMPARISON OF TERMINAL ALTERNATIVES
IN TERMS OF PERFORMANCE AND COST

User Requirements	Possible Points	Terminal A	Terminal B	Terminal C	Terminal D
Screen Size, Resolution	20	10	15	18	16
Product Reliability	20	15	15	12	10
Throughout Speed	20	14	14	16	18
Ease of Operation	30	25	22	26	24
Flexibility of Operation	30	28	26	24	22
Service Response Time	10	8	8	6	6
Service Repair Time	10	8	8	4	4
Quality of Manuals	20	12	14	16	14
Quality of Training	20	14	16	16	14
Ease of System Growth	20	12	14	16	16
Total Points	200	146	152	154	144
Cost (Purchase)		$3400	$3600	$3700	$3500
Cost Per Unit Needs		23.28	23.68	24.02	24.31

MODEMS & COMMUNICATION LINKS

A modem includes a modulator-demodulator and its interface to the digital equipment. The modulator converts binary data into a form suitable for transmission over a non-direct-current-coupled communication channel. The demodulator acts in reverse converting the signal back into binary form. The interface between the modem and digital equipment has been fairly well standardized throughout industry. Usually the voltages at the interface conform to an E.I.A. standard RS-232, which specifies bi-polar, baseband signals within a certain voltage range. In the simplest cases the interface contains leads devoted to outgoing and incoming data signals, and commands such as "clear to send," "data set ready," and "data terminal ready." These control leads enable the terminal to notify the modem when it is ready to transmit, and the modem in turn can notify the terminal when the connection is ready for date transmission. (Ref. 8)

Prior to the Carterfone ruling in 1968, only modems offered by American Telephone and Telegraph Company could be electrically connected to a switched network. A way around this tariff restriction was the acoustic-coupled modem, where a standard telephone headset was inserted into an acoustic interface connected to a data modem.

Due to the nonlinearity of the coupling mechanism the acoustic-coupled modem is limited to around 1200 bps.

This model continues to be popular due to its portability in teletypewriter applications even though electrical connections to switched networks is now allowed.

Modems operate in full duplex and half duplex modes. Full duplex allows simultaneous two-way channel communication, while half duplex only permits transmission in one-direction at any instant. Half duplex is normally used with low-speed CRT or typewriter terminals. Full duplex is used for these terminals when operating in the echo mode. These units are usually asynchronous character oriented. Full duplex is also used for synchronous block-oriented transmissions operating at higher rates.

Modems can also be clocked or non-clocked. A clocked communication channel is one where the receiving modem supplies a clocked signal to the digital interface for each transmitted data bit. The basic line rate is then controlled through a phase-lock-loop technique between the receiving modem and the transmitting modem. This gives these systems a greater data rate for a given line bandwidth, as compared to non-clocked systems.

Non-clocked systems are better suited for low-speed asynchronous terminals. The limitation is due to the usable data rate for a given line bandwidth.

There are several combinations and typical characterizations between clocked-nonclocked channels, and asynchronous/synchronous data formats which are depicted in Figure 3.

Modems may also operate in a parallel mode, where bits are transmitted as a single character simultaneously over a parallel, frequency multiplexed channel.

	Asynchronous	Synchronous
Non-clocked Links	- Low speed terminal communication - to 300 bps - character oriented	Seldom used
Clocked Links	- Medium speed - Terminals or buffered concentrators - 1200 to 9600 bps - character oriented	- medium speed - block transmission - buffered terminals and store and forward concentrators - 2400-9600 bps.

Figure 3

Various degrees of signal enhancement may be required for different modems over private lines. High speed modems use a technique called automatic equalization to alleviate signal distortion. Data rates are increased by using automatic equalization.

CONCENTRATORS AND FRONT END PROCESSORS (FEP)

There are three basic types of concentrators which will be discussed, nonbuffered concentrators, buffered concentrators, and store-and-forward concentrators.

NONBUFFERED

Non buffered concentrators, also called multiplexers, are used to interleave multiple low speed communication channels onto one or more higher speed channels for economy of

transmission (i.e., it is cheaper to multiplex some number of low speed lines onto one high speed line over a given distance than it is to string the same number of low speed lines over that distance). These concentrators are completely transparent in that no data or format modifications are possible.

BUFFERED

Buffered concentrators for the purpose of this discussion will be defined as those concentrators which buffer at the character level. These concentrators are not transparent since appreciable transmission delays are inserted. Each character is handled asynchronously, will start-stop bits indicating start and end of characters.

STORE-AND-FORWARD

These concentrators are capable of storing blocks of information. They are analagous to the use of a "selector data channel" on a computer where the data channel is dedicated to a specific device or function for the duration of the block transfer.

Store-and-Forward concentrators offer considerable flexibility. They become ideal points at which to modify or reformat the message. This is particularly attractive when multiple character sets and/or multiple transmission formats exist in a large network.

APPLICATIONS

Concentrators are used not only for remote concentrating (multiplexing low speed lines to high speed lines) but as front end processors. Virtually all front end concentrator

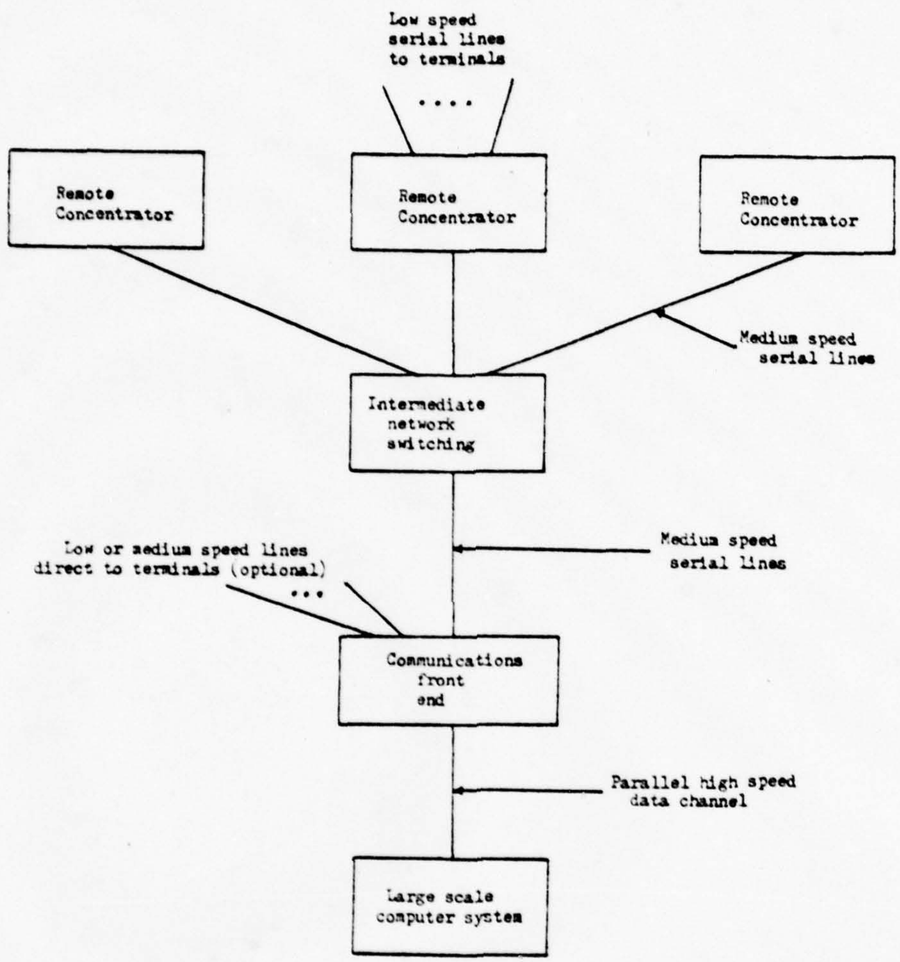


FIGURE 4a - Concentrator Applications

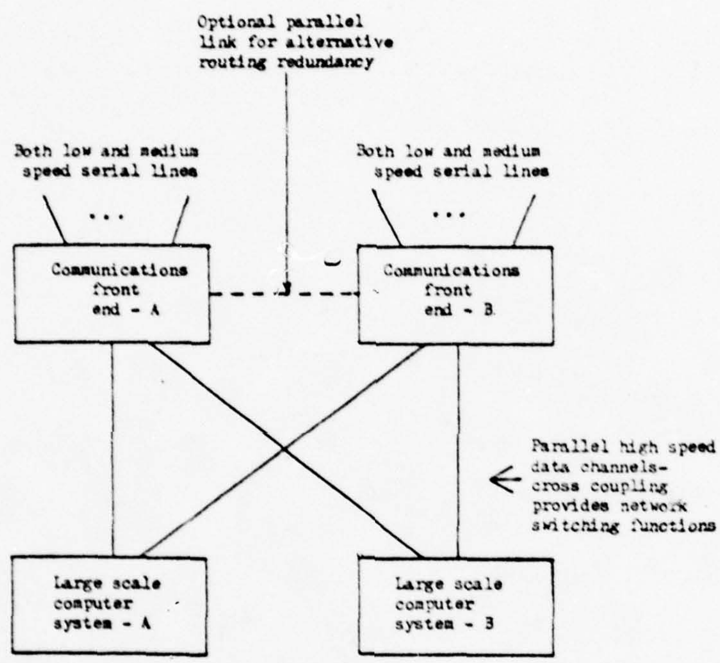


FIGURE 4b - Concentrator Applications

systems include a character-level interface. When the interface is supported by a small computer with buffered memory, the advantage of a message-buffering system is possible. (Ref. 8)

Large amounts of character-level input-output are costly, in terms of machine resources. The main frame is continually processing interrupts to service character-oriented communications reducing processing efficiency. It is desirable to maximize the amount of data transferred and processed via an I/O transaction.

There are several ways to separate input/output and main processing. A separate and dedicated front end processor is one approach. Another is a multi-processor system in which one or more processors handle I/O and lower level communication interfaces. The distinction between these approaches lies in implementation only not in the logical end result.

The use of a separate and dedicated FEP adds a degree of flexibility for large networks since the failure of a mainframe does not disable the network from rerouting incoming messages, if the FEP has network switching functions incorporated in it.

Two possible schemes utilizing concentrators are shown in Fig. 4a and b.

D. NETWORK DESIGN GOALS

The goal of a network designer is to select and configure the set of hardware and software elements that will provide the user with an optimum information collection, processing

and distribution capability. (Ref. 19)

In order for the designer to efficiently accomplish his goals he must keep in mind the motivational goals as well as the functional goals for the system.

MOTIVATIONAL GOALS

There are two sets of goals which should be considered in this category, the motivational goals of the user community as viewed by the users and the motivational goals of the system development as viewed by the system programmers.

These goals will be specified for a general purpose system although they may equally apply to special purpose systems. The lists do not attempt to list all goals which might exist for a system, only the basic goals.

For the user community, the basic goals of a general purpose computing facility include the following: (Ref. 8)

1. Capability - The system should be able to fulfill the needs of its user community, providing adequate performance (throughput, response time, etc.), computing and storage capacity, availability, and generality.
2. Evolvability - The system should be able to continue to fill the changing needs of its user community by graceful evolution. There should be long-term continuity of the user interface, including continuity of system commands, conventions, and standards.
3. Convenience of Use - A system should be easy to learn and easy to use. It should be well documented. It should provide a simple user interface, with ease of

program and data handling, and should be tolerant of user shortcomings. There should be no fundamental incompatibilities between interactive and non-interactive use, especially with respect to storage usage.

4. Reliability - If the system attempts to maintain information on line in a system-managed storage hierarchy, the continual availability of this information should be guaranateed. The hardware and software should operate with little or no malfunction. In case of any malfunctions, ill effects should be made invisible to users whenever possible.

5. Efficiency or Cost-effectiveness - The efficiency of the hardware and software is only a partial contributor to the cost-effectiveness. Optimization must be considered over the entire user community, examining the cost effectiveness of the system with respect to planning, designing, implementing, debugging, integrating, maintaining, managing, using, and evolving the system. Such global optimization is of course difficult to achieve. Furtermore, cost must be measured in various units, only some of which are monetary; the intangible costs of system unavailability, bad documentation, security violations, and system misuse are relevant but extremely difficult to evaluate.

Of course, not all of these goals may be optimized. There must be tradeoffs. A great deal of care and good judgement should be exercised during all stages of design, implementation,

integration, and evolution of the system in making these decisions.

For the system development group the goals are basically the same but from a different viewpoint. (Ref. 8)

1. Capability of the system adequate to support each stage of the development. It is highly advantageous to use a system as a development tool for its own development as soon as possible. In this way the use of the system is shaken down as a by-product of the development, and considerable experience can be gained. Difficulties tend to become visible sooner, and thus to be correctable sooner.
2. Evolvability of a system is at least as critical to system programmers as it is to users. No one is omniscient. Ideally a system should be designed in such a way that modifications and improvements can be accomplished gracefully.
3. Convenience of program development, program debugging, program interfacing, documenting, and maintaining is essential. An apparent detour in building a development tool, a debugging environment, can often be significantly rewarding in later stages of development. Convenience is enhanced by rigorously enforced standards and conventions and by automated design aids, such as languages suited to defining operating system functions.

4. Reliability is critical to system programmers, if the system is to provide a useful self-contained development environment. Its achievement depends on the entire development process, including the correctness of the original specifications of system goals.

5. Flexibility, especially dynamic flexibility, is particularly important. A system should be able to be highly reconfigurable, both at system startup and while in execution, operating with a wide variety of configurations as needed, and able to operate in spite of various component outages.

FUNCTIONAL GOALS

The functional goals or network functions must be identified independent of the specific applications that the network itself will accommodate. This approach provides a much greater degree of flexibility in that the same basic set of functions can be applied to any network regardless of its size or applications involved.

This set of functions can be organized in two basic groups -- those concerned with the processing and manipulation of the information to produce the desired results, and those concerned with the flow of information to and from the processing computers. These functions are called information processing and network processing. (Ref. 19)

The network design can be categorized into six levels:

Level 1 - Network Level

On this level the determination that a network is required is made.

Level 2 - Processing Level

On this level those functions related to processing information are separated from those related to data flow through the network.

Level 3 - MACRO Function Level

This level further separates the processing level so that appropriate software and hardware may be chosen for each function

Level 4 - MICRO Functional Level

The basic forms of hardware and software macro-functions are identified

Level 5 - Element Level

Specific forms of level 4 micro functions are selected.

Level 6 - Device/Technique Level

Specific hardware devices and/or software techniques are identified and selected.

The most logical point to begin the design approach is at the locations where data enters (sources) and data leaves (destination) the network. The data may pass thru relay points moving from source to destination (Fig. 5a). The information source and destination devices usually exist in some form of logical grouping called clusters. The designer should provide these clusters with a level of access to the network which

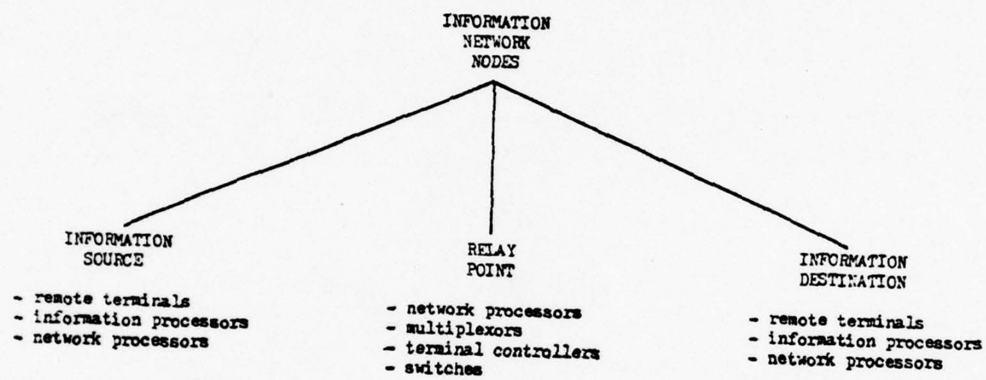


FIGURE 5a - Network Data Flow

allows the user to process his information within the desired response time.

Of equal importance is the task of assuring that the network elements are configured so as to provide the specified level of network availability. As a minimum this involves, among other techniques, the possible duplication of hardware and/or software, giving the system elements of redundancy.

Not all six levels will be discussed in the remainder of this section, but the framework of the approach will be presented.

Figures 5b and 5c provide the hardware and software functions for information processing.

Figure 5d identifies the five hardware functions required to configure any network from the smallest to the largest. The design of the network becomes the selection of appropriate subsets of these functions.

Figure 5e identifies the six basic software functions that are necessary to control hardware functions and data flow in the network.

The following definitions are presented to further define some of the functions at level 4. (Ref. 19) (Figures 5d, e)

Concentration

The concentration function is applied at relay nodes within a network to channel the flow of information between some number of source/destination devices in a cluster and some smaller number of lines or trunks connected to distant network nodes. Concentration occurs in two basic forms -- centralized and distributed multipoint.

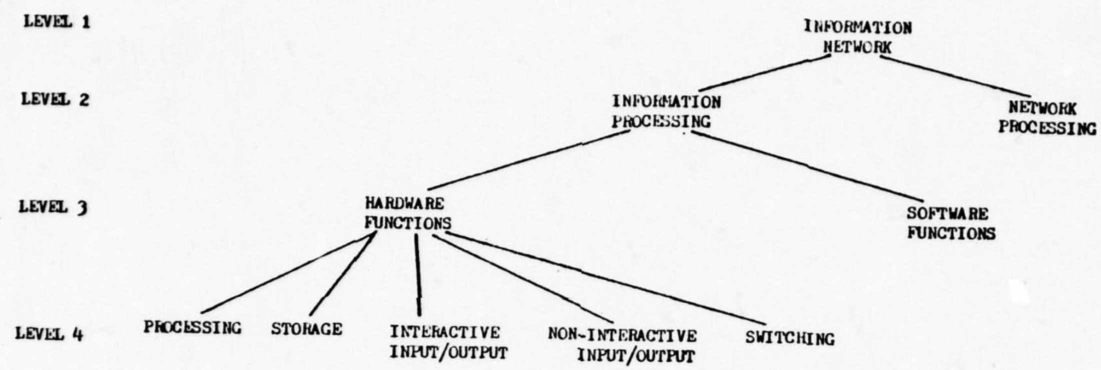


FIGURE 5b - Hardware Functions

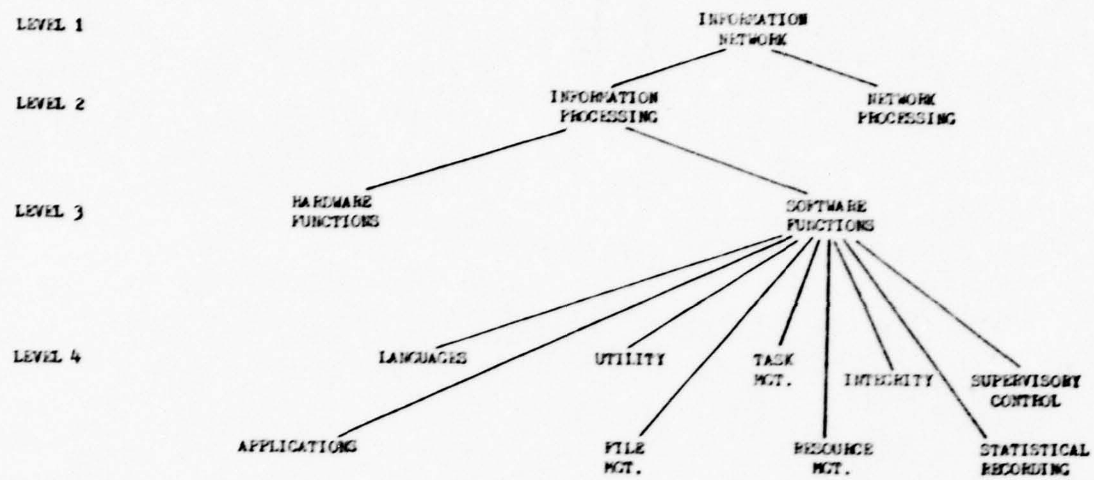


FIGURE 5c - Software Functions

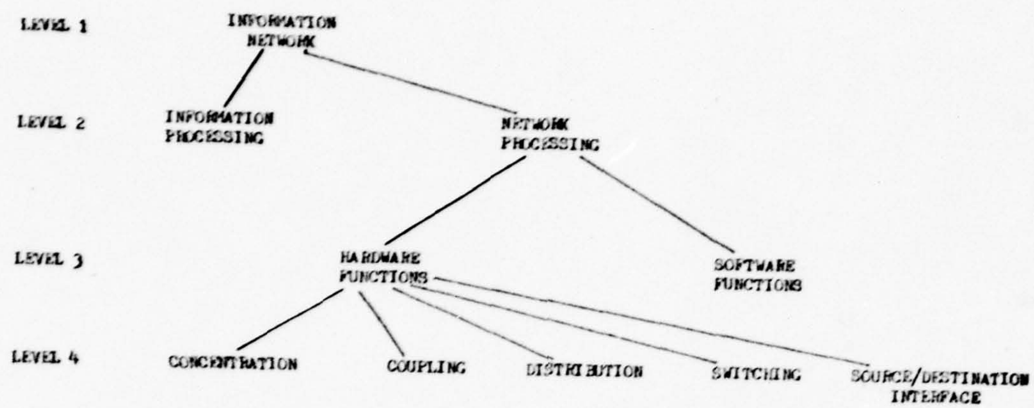


FIGURE 5d - Hardware Functions

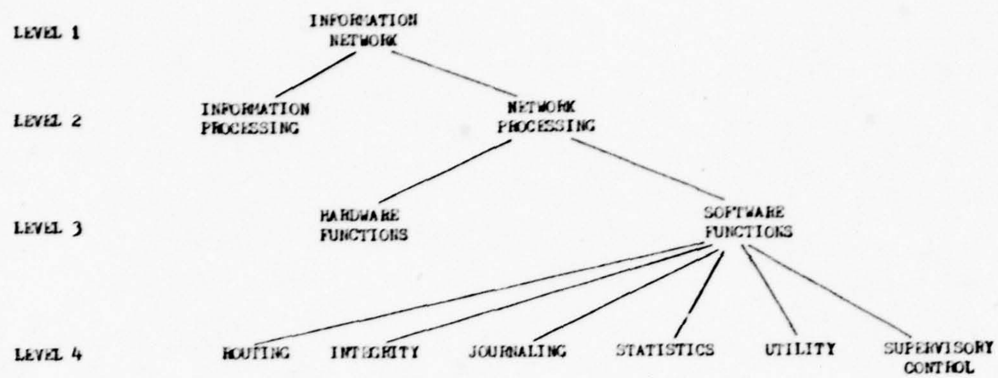


FIGURE 5e - Software Functions

Coupling

The coupling function provides an interface between the source/destination/relay devices of the network and the various lines and trunks they are connected to. Coupling devices convert the signals generated by the source/destination/relay devices into a form suitable for transmission on the lines and trunks when they are in the transmit mode and perform the reverse functions when they are receiving.

Distribution

The distribution network provides the medium or path over which information flows between nodes in the network. A large variety of lines and trunks are available, ranging in speed from less than a hundred to several hundred thousand bits per second.

The distribution network facilities fall into three basic categories -- in-plant, dedicated, and public switched.

Switching

The switching function provides a means for establishing and/or altering the path of information flowing through a relay node in an information network.

Source/Destination Interface

More important than organizing the source/destination devices in the network into various classes is recognition of operating characteristics that will have an effect on other network hardware and software elements. A partial list of these characteristics includes the following:

- Does the device operate in a continuous or intermittent mode?
- Is its location fixed, mobile, or protable?
- How is it coupled to the distribution network?
- Is it an information source or destination or both?
- Is it buffered or unbuffered?
- Is it a single speed device or capable of multiple speeds?
- Is it single or multiple code set oriented?
- What error detection/correction capabilities are included?
- How does it identify itself?
- Is it a programmable device or operator controlled?

Figure 5e identifies the six basic software functions that are necessary to control the hardware functions of Figure 5d and the flow of information in the network.

Routing

Routing includes those functions necessary to assure that the information entered into the network will be properly identified as to its source and will be directed to the specified destination(s) within the desired priority level and time frame. The routing function encompasses the following: addressing, trunk selection and routing, load leveling, priority control and queueing, reroute and intercept and the interface to the information processor(s) of the network.

Integrity

Network integrity includes those functions necessary to insure that the information is delivered accurately to the

proper destination, that hardware failures are detected and appropriate action taken, and that only those source/destination devices authorized will gain access to the network.

Journaling

The journaling function provides the network with a means for retaining copies of information flowing in the network. The journal can then be used for the retransmission of information and, of equal significance, as an aid in the network's restart/ recovery procedures. Network journals may be maintained at a single, central location or may be kept on a decentralized basis at several locations.

Statistics

The collection and evaluation of statistics concerning the flow of information in the network provides the designer with an insight to its performance characteristics, and is a valuable tool in maintaining an optimum configuration as growth and expansion occur. Depending on the size and geographic characteristics, the statistical recording function may be performed at a single, centrally located, site or at two or more distributed sites. Similar to the journaling function, the small to medium scale networks will maintain the statistical records at the centrally located site. Larger network will collect the statistics on a distributed basis and may periodically send them to a central location for inclusion in a total network summary report.

Utility

The utility function includes those tasks that must be

performed but are usually considered as overhead and, if possible, should be removed from the information processor and executed elsewhere in the network by the network processor(s). A partial list of utility functions includes format control, code translation and the execution of various supervisory control functions.

Supervisory Control

The supervisory control function provides an active interface between the operating network and the supervisory personnel. Smaller networks may require only a single supervisory control station while larger ones may have several which are organized in a hierarchical structure. A partial list of the supervisory control functions includes initiation of start and end-of-day procedures, intercept control, adding or removing terminals and/or lines to the configuration, changing routing tables, requesting a journal search or statistical report, initiating restart/recovery procedures, initiating network reconfiguration and others.

The structured approach to functional design provides a logical approach to designing a network. Because of its generality and application independence it gives the finished product flexibility and evolvability of design.

Any design constraints to the system may be plugged into the structure at the level and function where they are applicable. This approach will be used later to develop alternatives that will be presented for the NPS system.

II. COMPUTER PERFORMANCE MEASUREMENTS FOR TERMINAL ORIENTED SYSTEMS

There are two areas of concern for computer performance measurements for on-line systems, the man-machine interface, and the system performance. The first of these does not always receive the attention it warrants.

Since the user's view of the system ultimately determines the success of the system, performance and behavioral factors considered important to user groups should also be measured as a part of any system development. In addition, this type of user parameter should be continually measured as a part of any computer performance measurement program, for the reason that performance tuning certainly affects how the user's evaluation of system performance.

This chapter will look at three areas: user behavior in on-line systems, performance measurement tools for system and user behavior measurement, and finally guidelines for a user to start a performance measurement and evaluation program.

A. USER BEHAVIOR AND RESPONSE TIME

From a human factors point of view, several studies discuss response time requirements for interactive time sharing system interfaces. (Refs. 3, 4, 5, 6)

Response time or more specifically system response time was defined as the time between the last character input and the first meaningful character returned. Studies summarized in Ref. 3 indicate reponse times of less than 4 seconds are

generally sufficient to maintain continuity of dialogue for edit and file building type functions. Little additional benefit is found in reducing response below 2 seconds. Occasional responses requiring as long as 15 seconds are tolerable if the user expects beforehand that the system will have to do some significant amount of work to process certain transactions.

In general, response should be consistent for the same type transaction; large variations in response are disconerting to the user. (Ref. 3) This last notion is supported by another study which examined the proposition that increases in display rate would result in corresponding user performance increases. (Ref. 5)

In this particular study it was found that doubling the display rate from 1200 to 2400 bps produced no significant performance or user attitude changes. It was discovered that increasing the variability of the output display rate produced both significantly decreased user performance and a opinion of the system. Thus, while the display rate is certainly important (especially in real time environments), it affects user satisfaction and performance much less than an irregularity in display output rates.

For this reason, increasing output display rates (e.g., by buying faster terminals) should not be attempted unless it is discerned that the existing CPU can handle the increased data flow, thus guaranteeing consistency in the output display rate.

In another study (Ref. 4) it was found that system response time was relatively insensitive to variations in user think

times or typing rates. During the test, when users reported good response times, the data collected indicated that core memory utilization was approximately 80%. Increasing the number of terminal users on the system at this point produced very small additional gains in work being done, but resulted in longer average response times and erratic system behavior.

Operating under this load, it was found that the average response time was approximately 3 times the response time a single user on a dedicated system would see plus the delays required to load the users program into core.

Degradation of the system as more terminals were added tended to act like a step function. Up to a threshold of terminals there was no noticeable change in the system response time. Then, there would be a significant increase in response time which would hold for a few more terminals, then another step would occur. There were few steps in this function, with the later steps having response times of several minutes. (Ref. 4)

While the 80% utilization figure is surely machine dependent, it still suggests that there is a memory utilization figure where response time becomes excessive. This characterization was mentioned by Herndon who stated;

> "terminal response time is principally affected by the core size, which establishes its relative priority for loading into main memory." (Ref. 3).

A study conducted on the IBM System/360 Time Sharing System (TSS/360), the predecessor to TSO, indicated the importance of the user knowing how to use the commands. The results

seemed to indicate that a large number of users know and use only a few commands, or use only the simplest form of the commands. The result is that they often use lengthy sequences of commands to accomplish what a single command could do. Since almost any command language format can be implemented, behavioral criteria can be used as the basis for selecting formats that best suit users' needs and habits. (Ref. 6) Careful study of the behavior of users at specific geographical location and the use of this information in the selection of the command language for that site may return benefits in user performance many times the initial effort required.

In the NPS system no response time measurements were made. Nevertheless, it is the concensus of opinion that response time is unsatisfactory when more than a few users are on line. (Ref. 22).

B. SYSTEM MEASUREMENT TOOLS

The importance of computer system measurement and improvement has been recognized for some time. Some of the objectives in obtaining performance measurements are: to provide measurement capabilities for performance improvements, to describe current system performance and to provide a basis for evaluating decisions and future alternatives.

The vehicle for obtaining performance measurements has been the system monitor. There are four basic packages that have been traditionally used for system monitoring: the accounting package, the software monitor, the hardware monitor, simulation or some combination of these. In later years, and

designed specifically for on-line systems, two other types of performance measuring tools, merit discussion; these are remote terminal emulation (RTE) and Rand Monitor/Stimulus-generators (RMS).

ACCOUNTING PACKAGES

Accounting packages are included with almost any large system. Of course, the detail with which various data are collected is different from system to system. But nevertheless, it provides some advantages. (Ref. 13)

A. Advantages

1. exists in the system
2. identifies jobs resource utilization (CPU time, elapsed times, disk and tape accesses, data set activity)
3. usually low storage volume
4. usually reasonably accurate
5. identifies activity by job

Some disadvantages are:

1. records are difficult to work with
2. records are created independent of CPU state
3. 2 to 15% system overhead.

SOFTWARE MONITORS

Software monitors are programs which run consuming a portion of the system's assets (e.g., memory, CPU cycles, etc.). There are two types of software monitors; time driven and event driven.

The time driven monitors extract information from memory

at certain times. Only variables which can be expressed as a function of some total time can be measured. At each change in a variable of interest, its counter is incremented by the previous value of the variable multiplied by the time elapsed since the previous change. The counter is sampled periodically, and its increment divided by the elapsed time gives the average value of the variable. These type of monitors are usually easy to construct and require less overhead than the event-driven monitors.

The event-driven monitors extract information from memory at the occurrence of certain events (user defined). The data-gathering monitor may be implemented as part of the control program. The monitor is entered upon the occurrence of a timed interruption that is set for the required sampling period. The required data are moved to a buffer area, where they are some time later written onto tape or disk. With this monitor the number of occurrences of events can be registered without any problems, but for measuring the duration of the events the accuracy of the results depend on the hardware timing cycles used in the computer system.

Some advantages and disadvantages of software monitors are listed below; (Ref. 13)

A. Advantages

1. easy to use
2. portable
3. can monitor entire system and point out bottlenecks

4. relatively inexpensive ($8-12K) 1975 figures
5. can also monitor individiual programs
6. can determine queue lengths

B. Disadvantages

1. distortion of results (Heisenburg Uncertainty Principle)
2. high overhead while executing (10-40%)
3. operating system release dependent

HARDWARE MONITORS

When using a hardware monitor, small electronic devices (called sensors or probes) are attached to test points on the back panels of computer equipment. The sensors use a differential amplifier to detect voltage fluctuations at the locations to which they are attached. The voltage fluctuations represent such changes in status of computer components as busy/not busy, true/false, etc. The signal state, as detected, is transmitted by a cable from the sensor to the hardware monitor. These pulses may then be sampled and counted over a selected time interval. The reduction of this data by a software package can then produce analysis reports of desired performance parameters. (Ref. 10)

Some hardware monitor advantages and disadvantages are listed below; (Ref. 13)

A. Advantages

1. no CPU or device overhead on monitored system
2. no distortion of data
3. can sample or collect all events

4. extremely accurate

5. usually operating system and vendor independent

6. multiple CPU's can be monitored

7. operating system activity can be monitored

B. Disadvantages

1. relatively expensive ($20-70K) 1975 figures

2. talented users required

3. high setup time

4. can collect data, but provides little insight into meaning of data

SIMULATION

Simulation is dependent on many factors. First, it assumes the system can be accurately modeled. Secondly, in order to model it, a great many assumptions must be made. Nevertheless, it too has proved itself worthy of consideration. Some advantages and disadvantages are listed below; (Ref. 13)

A. Advantages

1. provides both performance and cost relationships

2. building models often uncovers bad designs

B. Disadvantages

1. expensive to use

2. high maintenance cost of models

3. more of an art than a science

4. difficult to validate the model

REMOTE TERMINAL EMULATION

Remote terminal emulation is an approach to the perform-

ance evaluation of teleprocessing systems in which a driver external to and independent of the system under test is connected to its communications device interfaces, either locally or through a communication network, and interacts with the system under test as if the driver were a set of terminal devices and operators. Integral to this technique is a monitor which captures data descriptive of driver/system under test interactions.

Scripts exercising various subsystems (compilers, editors, application packages, etc.) in scenarios that describe the user workload in a machine independent form are specified. All actions, pauses and decisions to be made by the emulated users are designated. A sample scenario might consist of;

Enter Program A
Submit program for compilation
Correct errors in lines 10 and 15
Submit program for compilation
Correct all remaining errors
Enter data "B" into file system
Executive program "A" using data "B".

The RTE's are implemented on various size machines from mini's to maxi's. The RTE's emulate not only terminal devices but also the human operators of those devices. Therefore, the human interactions with the system must be modeled. Two of the human characteristics which may be emulated and thus controlled are think time, and typing rate (or input rate). There are many vendor RTE's on the market. A partial listing is contained in Ref. 14.

The RTE approach to representing the variable workloads for interactive systems has proven itself to be a technically valid one.

RAND MONITOR/STIMULUS-GENERATOR (RMS)

The importance of response time in a terminal oriented system has been stressed in previous sections. This awareness prompted the development of a tool to aid analysts, the RAND Monitor/Stimulus-Generator. Mitre Corporation also produces a similar type of tool. The RMS interfaces between the terminal and communication lines; it stimulates the system by inserting a prestored message into the natural work stream repeatedly and measuring the resulting response times for the message. The message is usually a single command. By taking this simple approach, multiple samples of response time for an individual command enable analysts to determine the natural variability in system response. The times for individual responses can be added together to obtain script response times, and the variance computed, if required.

The critical assumption is that the performance of the on-line system in response to a command is context-independent. That is, the response time of a command is independent of preceding commands. This may not always be true, but the analyst may be able to assure that this assumption is valid by explicitly stating all conditions and then making sure they are met. (Ref. 12)

C. GETTING STARTED IN PERFORMANCE MONITORING

Vendors have many types of monitors available for their various systems. A typical listing can be seen for IBM in Ref. 10. It is hard to say which is best, each have their advantages and disadvantages, depending on the application. Much has been learned through many disasters which occurred when computer performance programs were started. (Ref. 11) A typical scenario suggests that a number of programs begin like:

1. choose a tool (a hardware or software monitor)
2. collect a lot of data with it
3. wonder what to do with all the data.

The basic principle should be to choose a small, easy to do, and cost efficient program from the large menu of computer activities. The following suggestions should be kept in mind: (Ref. 11)

1. limit your objectives at least at the beginning in order
   a) to understand system
   b) to make obvious improvements
   c) to do a) and b) in a cost efficient manner
2. concentrate on understanding the performance of your system
3. use continuous profile measurement; don't plan on massive, ad hoc tuning efforts during a crisis
4. employ local system programming talent in the performance program. Don't rely solely on outside consulting

5. use a small, bottom-of-the-line monitor
6. any changes to the system should include a before and after profile for comparison.

III. NPS CASE STUDY

A. THE PRESENT ENVIRONMENT

The W. R. Church Computer Center is an NPS organizational unit located on the first floor of Ingersoll Hall on the NPS campus. The purposes of the facility are to support faculty research and student instruction in modern computing methods, and to provide data processing support of administrative and library functions.

Since April 1967, it has accomplished these tasks using an IBM 360/67 computer. During this period, the requirements placed on the system have outgrown the capabilities of the system.

The Future Computer Planning Committee (FCPC) at NPS is currently formalizing plans for a replacement system. What that system will be is unknown.

Justification for the new system is partly based on two FCPC surveys and another independently taken. Conclusions from these surveys are summarized below. (Refs. 20, 21, 22)

Batch System

1. the present system has been operating for more than 12 years. Its failure rate in recent years has been unacceptable;
2. the school needs a computer at least ten times faster in processor and memory speed than the IBM 360/67;
3. a computer with at least 4 times the batch system memory capacity will be required.

4. graphics, terminals and a good data base system are needed
5. research progress is inhibited, grant opportunities are threatened and NPS research is becoming less competitive due to inadequate computing services.

Time Sharing System

1. time sharing usage has risen steadily since 1975;
2. presently, response time under time sharing is considered inadequate when more than a few terminals are in use
3. there is now a great demand for terminals even early in the quarter; students are standing in line to use terminals; the demand on weekends is almost as high;
4. sometimes it is not possible to use a terminal because of the inability to make a connection via phone company lines (public switched terminals);
5. the use of APL is intense;
6. professors are generating a greatly increasing time sharing load due to course work
7. there is need for communication between batch and time sharing systems;
8. there is a need for faster terminals;
9. Electrical Engineering, Aeronautical Engineering, Mathematics, and Operations Research departments all request availability of remote plotters so that graphics routines can be used interactively.

Whether or not all of these criticisms of the batch and time sharing system actually exist is inconsequential. The important point is that users think they exist. Thus, the computing environment on campus is basically one of user dissatisfaction.

In later sections of this chapter, the time sharing portion of the system will be analyzed in closer detail in an attempt to discover the possible causes of user dissatisfaction.

System Description

The present equipment, based on an IBM/360, includes two model 67 processors; four different levels of storage, including 2M bytes of core, 4M bytes on a drum, 24 disk drives with 29M bytes each and 8 disk drives with 100 M bytes each and 9 magnetic tape units; two high-speed plotters, fifty remote hard-copy and video terminals, and an IBM 2250 Graphical Display Unit. The two processors are identical and, by means of a configuration control unit, can access directly, or control, all components of the system including core storage modules, input/output controllers and devices.

The allocation of resources of the system to each CPU, using the configuration control unit is static and mutually exclusive; that is, one unit can only be used by just one CPU at a time; thus, the physical resources of the system are divided between the two CPU's.

The time sharing system normally operates according to the following schedule.

M, W	-	0930 to 2200
T, TH	-	0830 to 2200
F	-	0930 to 1800
Weekends	-	1300 to 1900

During these hours, the Computer Center operates with two independent systems, one CPU in the batch mode, the other under time sharing.

Batch System

The batch environment is a punched card oriented one in which jobs are submitted to a card reader (IBM 2501) located in Ingersoll Hall.

Batch jobs are assembled and run under IBM's OS/MVT release 21.8B operating system with HASP II extension to provide high performance input-output handling and job scheduling. Under OS/MVT/HASP, the system is capable of handling a variety of jobs written in different languages, using different amounts of central processor time, and various mixes of input-output devices.

There is little, if any, coupling between the two systems (batch and time-sharing); there is no way of automatically transferring one application program from one environment to the other. For example, if one edits and tests a program in the time-sharing system and wants to run it on the batch system, one has to first have the program punched and then has to submit the card deck to the batch processor.

The job scheduling at NPS is tuned to favor the short, simple, small jobs as depicted by the priority schedule below. The schedule decreases in priority from top to bottom.

JOB CLASS DEFINITIONS

CLASS	REGION	TIME	TAPES PER JOB
0	180K	20S	None
A	180K	20S	None
B	180K	2M	<3
C	250K	5M	<3
D	250K	5M	>2
E	350K	5M	<3
F	400K	30M	None
J	>400K	30M	None
K	>400K	30M	Any

FIFO in each class
Printing Priority is separate from execution priority and is based on actual lines of output generated.

Batch operations at the Church Computer Center support a variety of language processors. A listing is provided in Appendix A. (Ref. 1)

Applications packages available from the Center's library for batch processing are listed in Appendix B. (Ref. 1)

The public library stores programs in three forms:

a. Source programs - in high level language format

b. Object programs - compiled programs but not directly executable

c. Load programs - directly executable programs.

In addition to these, the computer center supports special user packages only accessible to one or a few private users.

Time-Sharing

CP/CMS is a time-sharing system developed for the IBM 360/67. The system consists of a control program (CP-67) with the Cambridge Monitor System (CMS).

The control program creates the time-sharing environment by supplying a virtual machine for each user while CMS

supplies the conversational or interactive mode in the form of a command language.

CP/CMS supports a variety of languages but not as extensive as batch. A list is available in Appendix C.

The CP/CMS public library consists of;

a. System library (SYSLIB) - a set of software facilities embedded in CP/CMS nucleus

b. SSPLIB - Fortran scientific routines

c. IMSLSP and IMSLDP - single and double point precision routines

The system allocates 8 Potter 4314 disc drives to CP/CMS, each containing 202 cylinders per drive. Each cylinder equates roughly to 1500 card images or approximately 120K bytes of storage.

Three of the drives contain directory information for USERID's, CP nucleus, paging and spooling space, T-disk (temporary work space) for users, CMS nucleus, and CMS libraries. The remaining five disc drives are used for general and private users of the time-sharing system.

There are two types of users on the system, private, and general. Private users request their own disc space where they can save files, whereas general users use system allocated areas for their programs (with no guarantee that their files will be saved).

The system allocates 66 cylinders for 33 general users. The remaining 944 are designated for private users. Usually these are allocated on the basis of 2 cylinders per user

although some may request and receive more than that.

When a user logs onto the system, he or she will automatically receive 256K bytes of virtual storage. On request, this may be increased up to 1024K bytes.

The CP/CMS system is configured in the following fashion: (also see Figure 6)

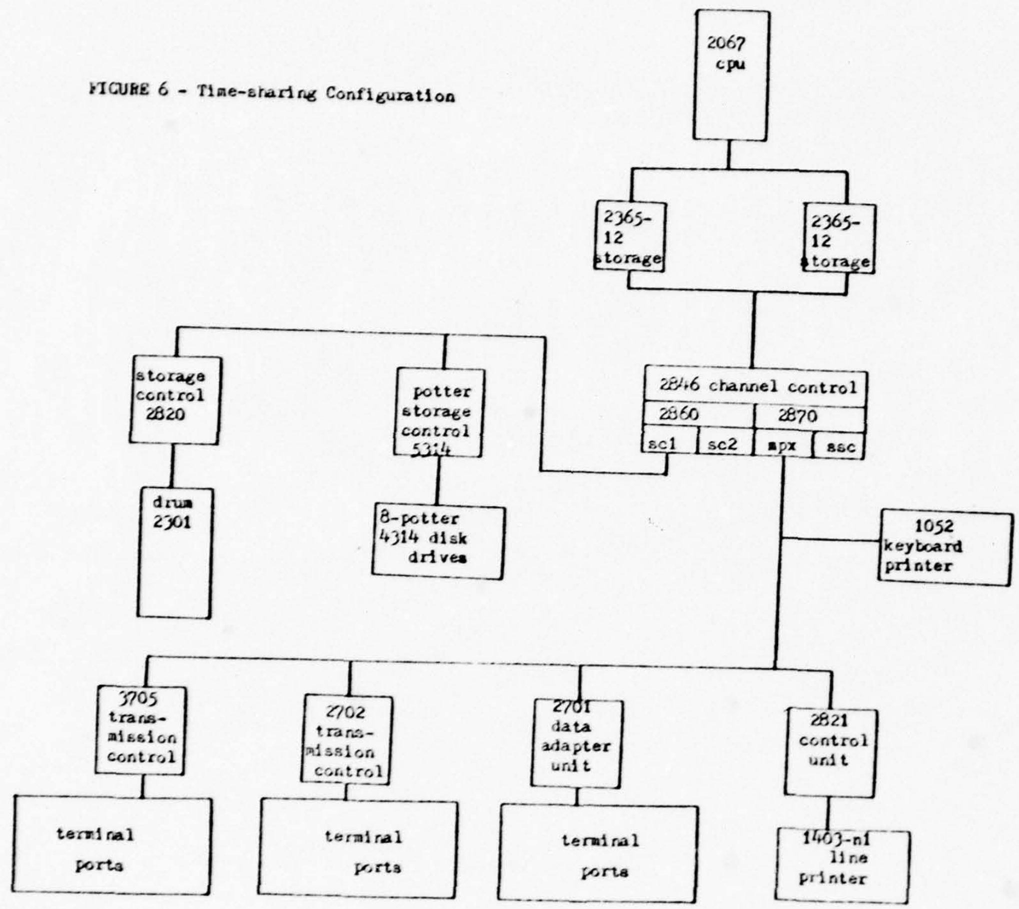
DEDICATED DEVICES

no.	Type
1 -	2067-2 processing unit
1 -	2860-2 selector channel
1 -	2870-1 multiplexor channel
1 -	2846-1 channel controller
2 -	2365-12 processor storage (256K bytes each)
1 -	2820-1 storage control (drum)
1 -	2301-1 drum storage (4M bytes)
1 -	5314 Potter disc control
8 -	4314 Potter disc drives (29M bytes each)
1 -	1052 keyboard CRT
1 -	2701 data adapter unit
1 -	2702-1 transmission control unit
1 -	3705 communication controller
1 -	1403-N1 line printer

In the event a Potter drive goes down or system user requirements change, the capability exists to add the following shared devices (devices which are normally on the batch side but can be reconfigured to the time-sharing side);

no.	Type	
1	2841-1	Storage Control (Disk)
2	2311-1	Disk Drives (7M bytes each)
1	2314	Control Unit
1	2314	Disk Drive (8 spindles at 29M bytes each)

FIGURE 6 - Time-sharing Configuration



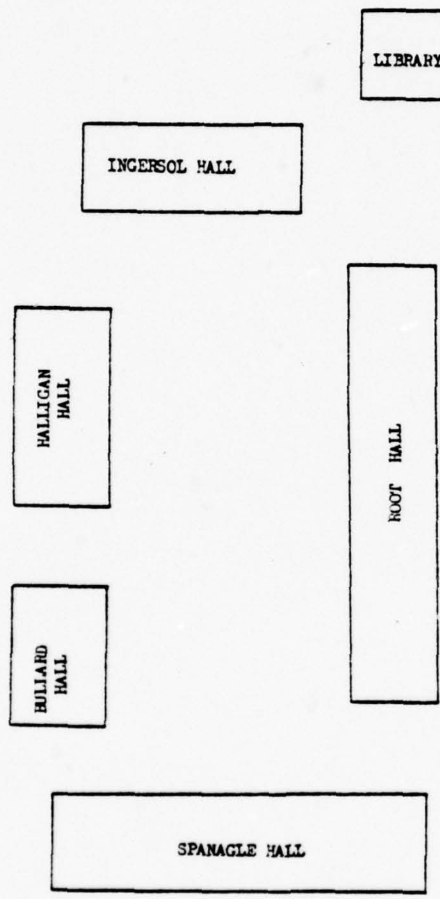


FIGURE 7 - NPS Campus

3420-7	Tape drives (320K bytes/second)
2402-1	Tape units (2 drives each)

There are 50 public terminals spread about the campus. Fig. 7 shows the topology of the terminal locations on campus and Table 2 lists the type, number, location, and modems used.

It is estimated that approximately 50 additional private terminals exist on and around the campus. It is not easy to account for the impact they have on the system; nevertheless, they cannot be ignored.

The NPS system allows any user with a valid user id number, project number, and terminal id number (any number between 1-99) dial up access to the system.

Since there are no RJE stations on campus, hard copy printouts from non typewriter terminals go to a dedicated line printer at Ingersoll Hall. Therefore, these users must leave their terminals to get a hard copy of their output. Even output for typewriter terminals may be printed at the Computer Center because the output rate is so slow that it may tie up a terminal for a considerable period.

The time-sharing system utilizes 3 transmission control units in its configuration, IBM 2701, IBM 2702, and an IBM 3705. Under the current configuration 59 I/O ports are supported. Table 3 shows the distribution of ports across the units.

B. USER BEHAVIOR ON CP/CMS

The time-sharing system at NPS is utilized by the student/faculty population. A survey taken of the major faculty users

TABLE 2

TERMINAL LOCATIONS

NO.	TYPE	MODEM TYPE	LOCATION
	DATAMEDIA		
1	Elite 1500	1	Sp-530
2		1	"
3		HW	In-151
4		HW	In-140
5		1	In-163
6	Elite 1520	HW	In-151
7		HW	"
8		HW	"
9		HW	In-152
	IBM 2741		
10		4	In-306
11		4	Ro-255
12		3	Sp-234
13		2	Sp-530
14		HW	In-136
15		HW	In-109
16		2	In-107
17		1	Ha-126
18		1	Sp-530
19		HW	In-102B
20		2	Ro-272
21		4	In-130
22		1	Ha-201C
23		4	In-354
24		1	Ha-205A
25		3	Bu-233
26		3	Ro-220
27		1	Library
28		4	In-354
29		1	Ha-247
30		1	E-309
31		HW	In-146
32		HW	In
	INTERTEC		
33		HW	In-149
34		HW	"
35		HW	"
36		HW	"
37		HW	"
38		HW	"
39		HW	"
40		HW	"
41		HW	"
42		HW	In-151
	LEAR-SIEGLER		
43		4	In-102A
44		4	In-133
45		4	He-M4A
46		4	Ha-245
	OMRON 8025AG		
47		1	In-108
48		1	Bu-102
49		4	Sp-544A
	TEKTRONIX 4012		
50		H/W	In-151

MODEMS

	Type	Model
1.	Anderson-Jacobson	A242A
2.	Data Systems (Livermore)	B
3.	Data Phone (Bell-Western Electric)	804B
4.	Anderson-Jacobson	ADC 260

HW - HARDWIRED

NOTE - This list is no longer valid. Many new terminals have been recently purchased since this survey and many have been moved.

TABLE 3

COMMUNICATION CONTROL UNITS

Unit	Number of Ports in Service	Access Mode	BPS
IBM 2701	2	Hard Wired	4800
	1	Patch Panel	4800
IBM 2702	10	Dial Up	134.5
	4	Dial Up	110
	5	Hard Wired	134.5
	6	Hard Wired	110
IBM 3705	2	RJE	9600
	1	RJE	4800
	5	Patch Panel	1200
	12	Dial Up	300
	4	Hard Wired	1200
	7	Hard Wired	300
TOTAL	59		

showed that they accounted for: (Ref. 22)

a. 7% of the total number of users;

b. 11% of the total number of sessions;

c. 16% of the total terminal time;

d. the average time per session of the major users is 1.4 times that of all users

e. only 37% of the major users polled have more than 256K bytes of disk storage.

These figures certainly suggest that the majority of time-sharing is indeed dedicated to educational student load rather than faculty research.

The data which will be analyzed in this section was gathered via the Computer Center CP/CMS utilization report. Only the data from Jul 77 to Jul 78 was used in the analysis.

Definitions of statistics that are not self-explanatory for Table 4 are included:

- Session: one logon to logoff period
- Total Logon Time: Summation of total time all terminals were logged on the system
- Avg Session Time: $\frac{\text{TOTAL LOGON TIME}}{\text{TOTAL NUMBER OF SESSIONS}}$
- Total CPU Time: Total Amount of CPU Time Used by all Users

A plot of the total number of sessions from Jul 77 to Jul 78 with the additional data for Aug, Sept and Oct is depicted in Fig. 8. These three extra months were included to show the quantum jump in number of sessions when the number

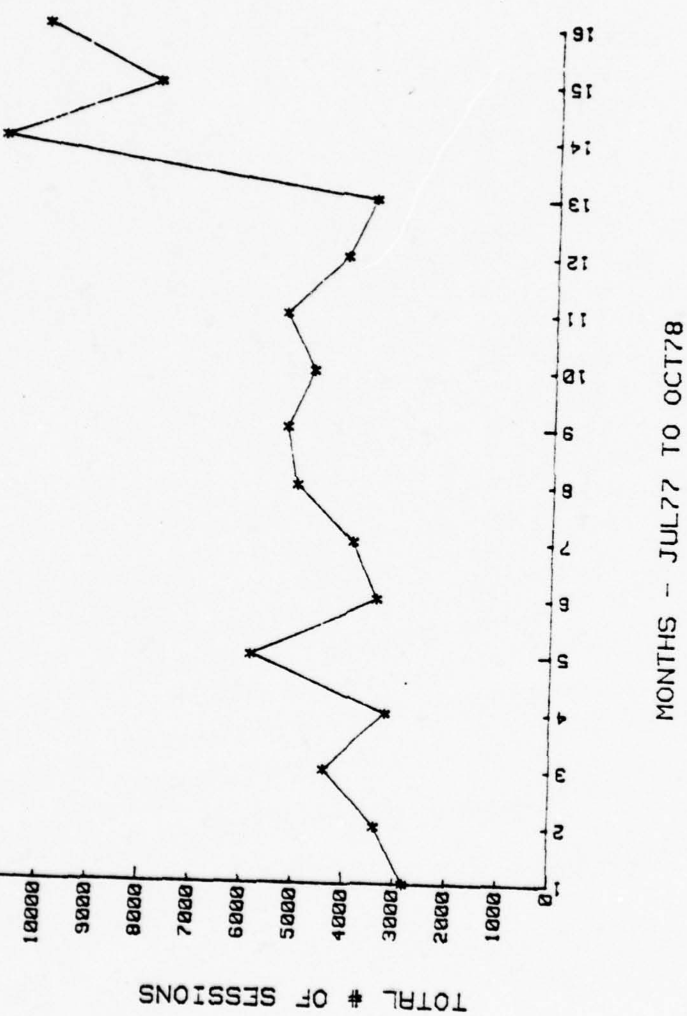


FIGURE 8 - TOTAL USER SESSIONS

TABLE 4
CP/CMS UTILIZATION DATA

MONTH/YR	TOTAL # OF SESSIONS	NUMBER OF SESSIONS FOR		LOGON TIME (HRS) FOR		CPU TIME (SEC) FOR		AVERAGE TIME (MIN) PER SESSION		AVERAGE CPU TIME (SEC) PER SESSION		AVERAGE SESSIONS PER USER	
		P	G	P	G	P	G	P	G	P	G	P	G
Jul/77	2768	2656	223	1899	101	189813	10140	42.9	27.20	71.46	45.47	17.24	4.37
Aug/77	3514	2959	555	1945	265	186462	16905	39.44	28.64	63.01	30.45	18.04	6.09
Sep/77	4836	2129	2707	-	-	-	-	40.5	2.85	-	-	15.31	43.6
Oct/77	3299	2651	648	1572	279	183526	14721	35.57	25.83	69.22	22.7	14.89	5.94
Nov/77	5727	4895	832	3417	441	291003	21671	41.88	31.80	59.45	26.04	23.42	5.86
Dec/77	3314	2925	389	2167	179	327876	5988	44.45	27.60	112.09	15.39	14.77	4.74
Jan/78	3828	2886	942	1731	430	131154	11009	35.98	27.38	45.44	11.68	14.21	7.47
Feb/78	4910	4151	759	2651	355	277481	60853	38.31	85.54	66.84	80.17	16.67	6.27
Mar/78	5194	4469	725	3755	388	342972	30373	50.41	32.11	76.74	41.89	18.31	5.57
Apr/78	4658	4176	482	2976	243	364407	12529	42.75	30.24	87.26	25.99	16.90	4.38
May/78	5200	4738	462	3378	241	441798	9679	42.76	31.29	93.24	20.95	19.26	5.77
Jun/78	4018	3744	274	3477	201	623652	12457	55.72	44.01	166.57	45.46	16.20	4.98
Jul/78	3552	3220	332	2982	162	575472	7523	53.88	29.27	178.71	22.65	13.58	4.58

P - Private Users

G - General Users

of usable terminal logon numbers was increased from 50 to 99. This jump occurred even though the summarization report is only designed to report data on valid logon numbers (1-50), which excluded an estimated 20% of data inputs. No justification for the increase could be specifically identified. Computer Center personnel could only comment that anytime access to the system is made easier, a corresponding increase in usage results.

The rest of the data will be presented in the following manner: student population vs user population, usage trends during 11 week quarters, daily usage trends, session characteristics, time sharing utilization factor, and language processor usage.

Student Population vs User Population

The student population at NPS from Jul 77 to Jul 78, except for a rise in Oct 77, has remained essentially stable. The table below contrasts the student population each month with the number of different users of time sharing each month.

Student Population		User Population
Jul 77	821	184
Aug 77	821	225
Sep 77	821	180
Oct 77	1018	248
Nov 77	1018	307
Dec 77	1018	255
Jan 78	991	284
Feb 78	991	336
Mar 78	991	335
Apr 78	971	317
May 78	971	301
Jun 78	971	271
Jul 78	998	287

October 77 signified the new beginning of the fiscal year, and with it came the funds to support the influx of students which had been slowed since July 77. This is why a sharp increase in student population is seen for Oct 77. If the projected increase in students of approximately 200 is realized, the system should expect a corresponding increase in user population of about the same magnitude that occurred in Oct 77. This would increase the user population by approximately 50, which is a significant increase.

Quarterly Usage Trends

Four quarters, Summer 77, Fall 77, Winter 78 and Spring 78, were observed. Figure 9 shows each of these quarters over its eleven week cycle versus the average number of users during each week. This last number was computed by taking the average number of users for each day and averaging them together for each week of the quarter to obtain the average number of users per week. This explains why the numbers might seem lower than expected. The general trend is as expected, usage is lower in the first few weeks and generally builds to a peak in the 10th week when most course projects are due.

If the 10th week which is the peak usage point for each of the quarters analyzed is isolated, an increase in the average of number of users per week can be seen:

QUARTER	Average Number of Users for Week 10 of the Quarter
Summer 77	7.9
Fall 77	12.5
Winter 77	14.2
Spring 78	14.4

FIGURE 9 - QUARTERLY USER TRENDS

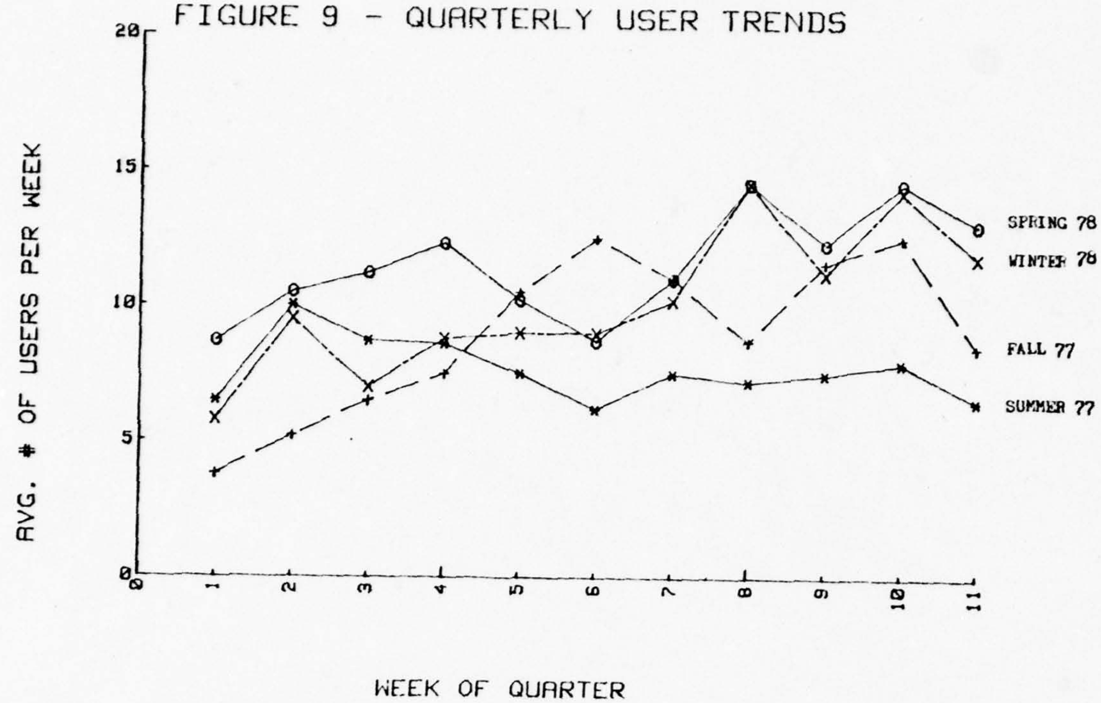
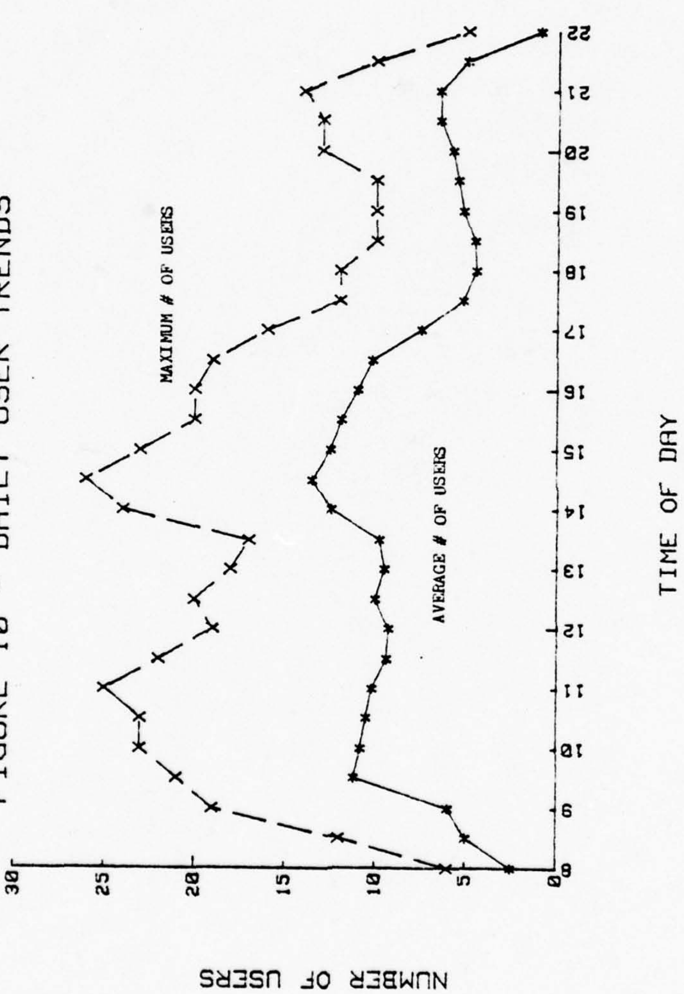


FIGURE 10 - DAILY USER TRENDS



Although relatively few quarters were sampled, it at least suggests an upward trend in time sharing usage.

Daily Usage Trends

Although usage trends on a monthly and quarterly basis vary from month to month and quarter to quarter, daily trends are much more predictable. The daily characteristics did not vary significantly enough in the data to warrant special analysis, so a typical day during the week (since these are of more interest than weekend days) was chosen from Jul 78.

The average of maximum and average of average number of users are plotted against time of day. As can be seen in Fig. 10 there are three maximums; roughly 0930 to 1030, 1400 to 1500 and 2030 to 2130. The two distinct valleys in the graph occur at lunch and dinner hours. Most classes are scheduled in the morning hours which explains the highest peak during afternoon hours.

Session Characteristics

When Table 4 was analyzed for session characteristics several observations were made. These are summarized below;

- Private users (those with their own disk space) tend to dominate general users in number of sessions by a factor of approximately 5 to 1.
- Each private user has more sessions than the general user by a factor of approximately 3 to 1.
- Private users sessions are longer by about 1.4 times that of general users.

- Private users use more CPU time per session than general users by nearly a factor of 3 to 1.

Time Sharing Utilization Factor

After analyzing all of the data, no single statistic could be found that measured time-sharing utilization; so using the data available and some facts about hardware usage, a utilization factor was defined.

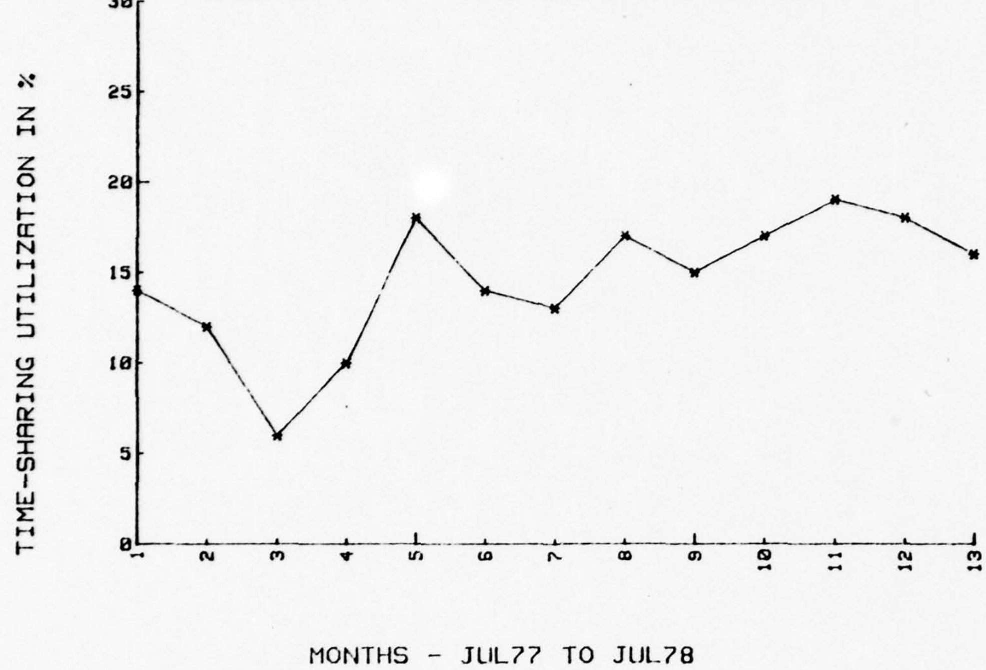
If the total connect time could be compared to the total available connect time, this would give a reasonable prediction of actual usage trends. After reviewing the data again it was decided that these two factors could indeed be computed.

The total connect time was derived from multiplying the total number of sessions per month by the average time per session. The total available connect time was computed based on the total number of I/O ports available (59) and the actual number of hours the system was up and operating each month. This number varies somewhat due to maintenance down time during normal operating hours but measures exactly the availability of the system. The ratio of these two figures was called the time-sharing utilization factor and was computed by the following formula;

$$\text{time sharing utilization factor} = \frac{\text{Total connect time}}{\text{Total available connect time}}$$

Figure (11) shows the results of this computation plotted against time. Because the observation period was only one year long, no statistical trends could be seen. The signif-

FIGURE 11 - TIME-SHARING USAGE



icance of the utilization factor, as presented, is that it gives a realistic picture of the actual utilization of the system.

Language Processor Usage

There were no quantitative figures for language utilization available nor were there any attempts to gather statistics on these factors. It must be pointed out though that language usage would be an important factor in the analysis.

In order to get any real picture of system requirements, benchmark tests under controlled conditions need to be carried out to learn the characteristics of NPS actual workload. This was not only beyond the scope of this thesis but was discouraged by the Computer Center due to possible generalizations from a probably less than representative picture of actual conditions.

On a qualitative note, it seems that there is a strong feeling that intense APL usage is a major factor in the NPS system. (Ref. 22) The increased usage of this language warrants specific analysis in this area.

As for other languages, Fortran is the most highly used language. Since the student population comprises the biggest group of users, the usages of all languages tends to follow the courses being taught in a specific quarter and the student load in those classes.

C. ANALYSIS PROCEDURE

The technology of system performance measurement is sufficiently well developed so that one is generally able to gather

unlimited quantities of data on almost any aspect of a system's operation. What is often lacking is a straightforward way of answering specific questions relating to system performance.

Classical statistical designs for experiments often lead to regression analyses and analysis of variance. The performance questions that can be answered using such techniques are of the type, "Is the performance of several different systems the same (are the measurements of performance identical from a statistical point of view)?" or "How does one variable depend upon the others?"(Ref. 7)

Bard in Reference 2 suggested a data reduction technique that was ultimately chosen as a model for analysis of data for the NPS CP/CMS system.

The CP-monitor that was available at NPS was a software monitor that runs in a virtual machine. The CP 67 system permits a privileged virtual machine to read the contents of specific locations in the control program's address space. The virtual machine "puts itself to sleep" pending the arrival of a timer interruption set by the user. By using these facilities, the virtual machine may sample the system counters at specific intervals. Each time it wakes up, it records the required data into a disk storage area. The sampling period can only be approximately regulated and therefore will vary slightly. Observations taken by this method will be biased by the fact that the measuring virtual machine must be in the run queue and executing at the time the observations are taken. The overhead that is imposed on the system is variable

and difficult to account for. And, perhaps worst of all, the monitor itself becomes part of the workload that it attempts to measure. However, the virtual machine monitor also offers some advantages: it requires no changes to the control program, it consumes no real storage space when not running, and it may be written in a high-level language (except for a simple interface routine). (Ref. 2)

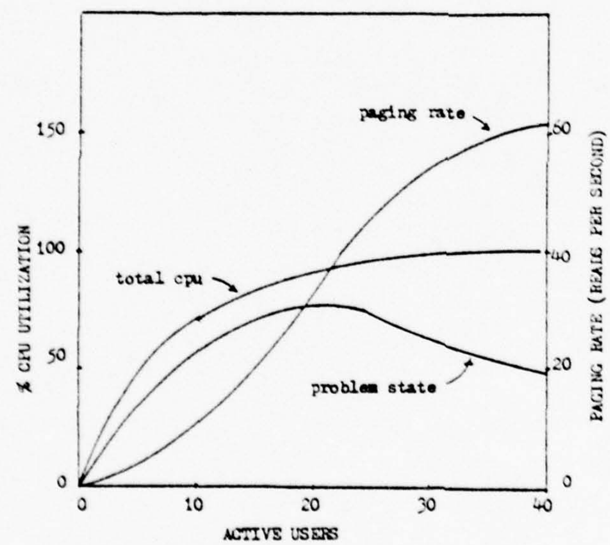
There are four major components in the VM/370 system that are also present in the CP 67 system; the CPU, main storage, the paging subsystem; and the I/O (other than paging) subsystem. Each of these areas are discussed below in terms of saturing the system and causing a possible bottleneck.

CPU

The CPU is saturated when its utilization approaches 100 percent. A truly saturated CPU can be cured only by being replaced with a faster one. However, some further analysis may reveal different underlying causes for the saturation and suggest cures of a less drastic nature.

One case in point occurs when the CPU becomes saturated with overhead due to paging. The case is shown in Fig. 12: total CPU utilization approaches 100 percent, problem state time declines, and paging rate climbs as the number of users increases. Such conditions prevail if, for some reason, the scheduler consistently underestimates working set requirements and thus maintains too high a multiprogramming-level (MPL). Reducing MPL will release some of the CPU time spent paging, but whether or not the remaining MPL will be sufficiently

FIGURE 12 - CPU saturated with paging overhead



high to maintain good throughput depends on the amount of storage available. Increasing storage capacity while retaining the same MPL would also decrease the paging rate and release some CPU time for productive use. (Ref. 2)

Main Storage

From the schedulers point of view, main storage appears to be saturated when the eligible to run list is almost never empty. Nevertheless, a saturated memory is not necessarily a performance bottleneck. If paging is moderate and the CPU is fully utilized, then main storage capacity is adequate and will have to be increased only after a more powerful CPU is installed. If both paging and CPU utilization are light, then the scheduler is probably overestimating working set requirements and consequently maintaining too low an MPL. If the paging rate is high, productive CPU utilization (percent problem state time) is low, and the MPL is high, then the scheduler may be at fault. This so-called thrashing condition may be removed by inducing the scheduler to maintain a lower MPL. Only if MPL is low, paging is heavy, CPU problem state utilization is low, is the saturated main storage a true bottleneck and in need of expansion.

Generally, acceptable performance is achieved if storage is expanded only to the point where an adequate MPL can be maintained. However, additional performance improvements are attainable by further increases of storage capacity above the saturation point. If more storage capacity is installed, then a substantial number of pages belonging to interactive

users can be held over from one interaction to the next. The total paging rate is thereby reduced, and CPU time previously spent on paging overhead is freed for productive use. Furthermore, response time to interactive tasks is improved. However, as the number of users on the system increases, the amount of excess storage required to hold temporarily inactive working sets increases proportionally. (Ref. 2)

Paging Subsystem

The following definitions are provided to clarify discussion in the remaining sections:

idle wait time -	CPU wait time when no high-speed I/O requests are outstanding
page wait time -	CPU wait time when outstanding I/O requests are primarily for paging
I/O wait time -	CPU wait time when outstanding I/O requests are not primarily for paging
total wait time-	idle wait + page wait + I/O wait

If page wait time accounts for the major part of total wait time (see Fig. 13), then the paging subsystem is probably at fault.

Page wait may be experienced either because the paging rate is too high or because page transit time is too long. The first condition, caused by working set requirement being underestimated by the scheduler, has been dealt with above. The second condition occurs when either no high-speed paging devices are installed or their capacity has been exceeded. The system may normally page to a fixed-head storage device, e.g., IBM's 2301 drum storage device. But when the number of

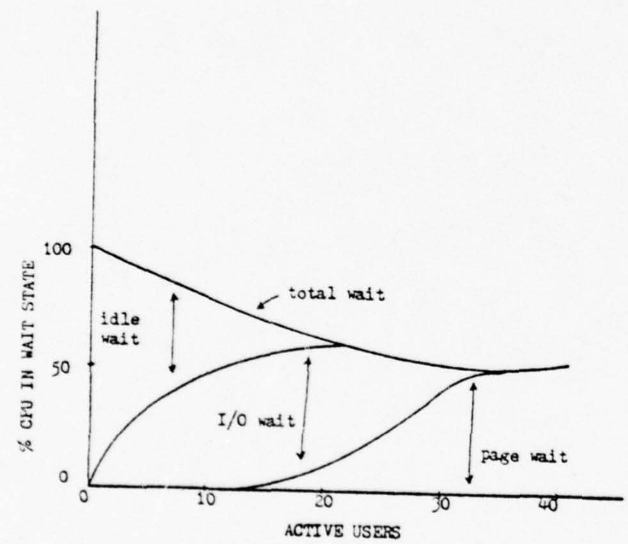


FIGURE 13 - Paging-bound system

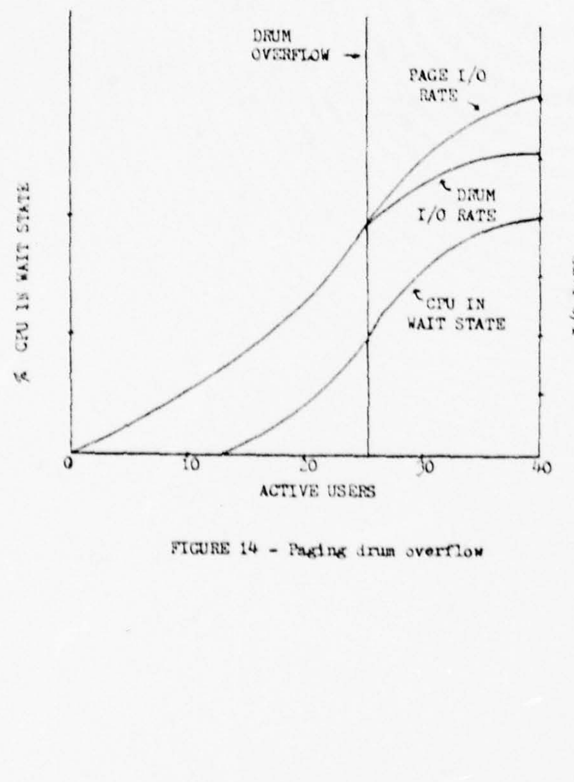


FIGURE 14 - Paging drum overflow

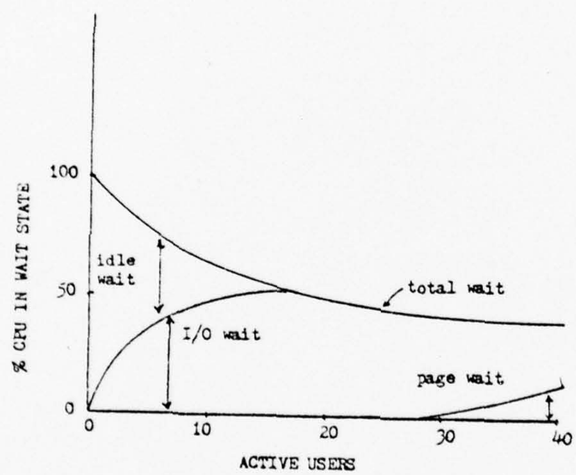


FIGURE 15 - I/O - bound system

users is sufficiently large, the drum overflows and some paging will be to the slower device. The point at which overflow becomes significant can be determined from a plot of total page I/O rate and drum I/O rate versus number of users. (Fig. 14). Various remedies short of installing an additional drum may apply. These are: free virtual pages that are no longer needed; reduce the size of user virtual machines; reprogram to use less virtual storage; increase use of shared systems. (Ref. 2)

I/O Subsystem

A bottleneck in the I/O subsystem reveals itself in a manner analogous to the paging subsystem. If there is enough main storage to maintain an adequate MPL, and yet I/O wait time is high (Fig. 15), a deficient I/O subsystem is indicated. It may be simply that the work load is so I/O-bound that no feasible expansion of the I/O facilities will handle it. Some rearrangement and/or expansion of the I/O subsystem will cure the problem. It will be necessary to measure the utilizations and the I/O rates of the individual I/O channels and devices. Then, better-balanced loading can be achieved by moving physical packs from one channel to another. (Ref. 2)

D. DATA COLLECTION

The foregoing analysis requires that performance variables be measured over a certain time period, say, one week of routine operation. (Ref. 2) Not all of the variables that are required for a complete analysis were available. The ones which were are listed below;

CPTIME	= CPU TIME IN SUPERVISOR STATE
PROBT	= CPU TIME IN PROBLEM STATE
WAITT	= CPU TIME IN WAIT STATE
OVERHD	= SUPERVISOR TIME NOT CHARGED TO USERS
WTIDLE	= WAIT TIME FROM PERIODS GREATER THAN OR EQUAL TO 1/4 SECOND
WTPAGE	= TIME SPENT WAITING FOR A PAGE
	* TIMES IN HUNDREDTHS OF SECONDS
PGEXCP	= COUNT OF PAGING EXCEPTIONS
PGREAD	= PAGES READ IN
PG SWAP	= PAGE SWAPS
PG STEAL	= COUNTER, USER IN QUEUE LOST PAGE
VSIOU	= USER SIO's
VSIOCP	= CP SIO's
VMIOU	= USER MULTIPLEX SIO's
USERS	= NUMBER OF USERS LOGGED IN

Data was taken for all of these variables over a 5 day period. Each day the monitor was started in the morning and allowed to run until the system "crashed," or allocaged storage for the program filled. A 30 second interval sampling period was chosen. Listed below are the time periods data was taken:

	time run began AM		time run ended PM
13 Nov Mon	8:36	to	16:50
14 Nov Tue	8:42	to	14:23
15 Nov Wed	9:09	to	19:25
16 Nov Thur	8:58	to	19:08
17 Nov Fri	8:39	to	15:35

The data was analyzed from two approaches;

approach 1: the data was combined for all five days and sorted by number of users.

approach 2: the data was sorted by number of users for each of the five days, giving five sets of data for comparison.

For both approaches, the data was partitioned into groups so that no fewer than 50 observations were taken for each group. The mean and standard deviation of each performance variable within each group of observations were computed using the BMDP (Biological Medical Program) statistical package. Then, for each variable of interest, a plot of the mean, against the number of users was drawn to check for any characteristics in the performance data that parallel cases discussed in the last section.

Table 5 lists the data of interest gathered via approach 1 and Tables 6a, b, c, d, e list the data for approach 2.

A dilemma arose between definitions used in the CP monitor and the definitions mentioned earlier for idle wait and I/O wait time. I/O wait time is not addressed in the CP Monitor program (Ref. 15) and idle wait time definitions differences are unclear.

Parameters which coincide between the CP Monitor and the definitions presented earlier follow:

a. page wait time equates to WTPAGE

b. total wait time equates to WAITT

c. CPU in problem state equates to PROBT

d. page I/O rate equates to PGREAD/time interval observed.

TABLE 5

* IN 100ths OF SECS.

CP MONITOR DATA FOR APPROACH 1

USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	PGEXCP	PGREAD	VSIOU	FMIOU	SAMPLE SIZE
9 or less	274.00	1140.77	1618.30	35.08	21.02	17.89	28.8	142.0	86
10	288.58	1616.90	1138.76	57.97	62.54	61.23	41.1	50.3	65
11	252.27	1320.37	1463.00	68.51	38.92	37.56	6.1	56.6	85
12	260.19	661.33	2121.24	103.06	57.99	57.8	20.4	72.3	115
13	318.78	800.96	1929.90	177.65	95.24	96.17	13.0	157.4	160
14	334.51	564.57	2152.84	213.73	99.74	99.05	12.7	182.2	140
15	352.52	1037.24	1672.71	355.91	180.21	184.30	13.0	189.3	85
16	331.59	1167.78	1553.29	221.34	130.63	138.46	11.8	164.6	102
17	388.51	915.89	1751.09	378.37	181.82	187.11	29.6	196.0	136
18	390.09	855.61	1812.53	295.58	170.65	181.62	35.8	201.75	251
19	405.55	989.29	1672.98	483.19	254.75	269.8	30.67	176.42	368
20	457.20	949.98	1681.16	518.81	321.69	344.17	31.76	208.33	282
21	464.20	1159.86	1482.94	516.33	323.00	347.96	25.48	230.41	366
22	515.59	1164.90	1429.61	647.67	390.23	422.15	30.37	248.49	453
23	523.04	1108.22	1477.18	645.10	390.23	424.64	37.38	229.43	280
24	555.90	1304.76	1295.91	806.84	516.28	563.01	27.63	262.84	330
25	569.775	1186.41	1417.38	787.68	487.95	532.87	31.07	295.47	376
26	622.30	1386.99	1214.87	855.95	599.86	656.14	*31.20	347.74	285
27	606.08	1181.81	1395.72	904.00	570.03	624.68	36.66	318.29	184
28	656.73	1274.14	1336.63	1124.75	705.75	770.67	34.60	255.98	181
29	690.13	1264.43	1382.06	1301.72	790.13	861.27	24.58	297.70	106
30	661.36	1395.48	1267.80	1196.94	728.86	803.07	23.47	191.26	76
31-33	676.56	1352.84	1206.42	1172.28	776.94	857.85	19.31	190.01	71
34-36	1049.89	844.76	1601.68	1556.89	958.69	1046.12	20.42	224.39	69
37-39	1037.01	1001.51	1343.35	1330.44	932.42	1031.8	18.11	193.91	68

. Sample size - the number of 30 second intervals observed

. All categories except sample size and users are computed means

TABLE 6a

CP MONITOR DATA FOR APPROACH 2 - MONDAY

*UNITS IN 100th's of SECONDS

# USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	*PGEXCP	PGREAD	VSIOU	VMIOU	SAMPLE SIZE
1-4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5-8	277.74	115.26	2634.63	41.05	13.474	14.68	58.05	66.32	19
9-12	301.05	664.75	2094.80	217.70	141.2	143.95	69.95	33.15	20
13-16	367.60	659.33	2094.73	534.39	281.86	285.19	26.13	77.86	15
17-20	420.94	689.24	1978.64	504.82	262.58	282.55	30.13	195.93	97
21-24	508.70	1064.48	1593.89	841.29	460.07	501.56	23.82	172.37	318
25-28	613.17	1036.54	1594.59	1252.17	671.13	730.72	35.41	241.87	210
29-32	763.87	992.31	1561.75	1422.66	764.21	870.73	26.19	210.58	79
33-36	1043.78	855.02	1594.38	1551.97	954.06	1041.34	20.49	226.81	73
37-40	1037.01	1001.51	1343.35	1330.44	932.42	1031.80	18.12	193.91	68

. Sample size - the number of 30 second intervals observed

. All categories except # Users and Sample Size are computed means

TABLE 6b

CP MONITOR DATA FOR APPROACH 2 - TUESDAY

*UNITS IN 100th's of
SECONDS

# USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	PGEXCP	PGREAD	VSIOU	VMIOU	SAMPLE SIZE
1-4	0	0	0	0	0	0	0	0	0
5-8	0	0	0	0	0	0	0	0	0
9-12	0	0	0	0	0	0	0	0	0
13-16	0	0	0	0	0	0	0	0	0
17-20	457.10	1117.31	1505.52	507.93	340.19	372.95	29.61	259.41	156
21-24	505.60	1235.67	1423.31	827.21	514.72	560.29	20.80	191.29	178
25-28	520.20	1034.25	1557.95	623.08	362.04	400.35	28.33	352.49	196
29-32	572.02	1478.21	1238.65	1198.34	767.35	840.41	18.44	188.85	119
33-36	0	0	0	0	0	0	0	0	0
37-40	0	0	0	0	0	0	0	0	0

. Sample size - the number of 30 second intervals observed

. All categories except #Users and Sample Size are computed means

TABLE 6c

CP MONITOR DATA FOR APPROACH 2 - WEDNESDAY

*UNITS IN 100th's OF SECONDS

# USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	PGEXCP	PGREAD	VSIOU	FMIOU	SAMPLE SIZE
1-4	0	0	0	0	0	0	0	0	0
5-8	245.66	105.83	2678.50	35.00	7.33	5.83	11.00	139.83	6
9-12	274.21	1585.73	1179.57	54.19	43.34	40.74	15.55	89.54	214
13-16	350.63	1126.53	1577.96	256.01	133.30	136.16	13.58	196.19	182
17-20	395.45	841.76	1828.73	566.83	273.02	285.61	21.73	155.02	345
21-24	535.53	1168.78	1406.87	901.26	507.48	546.31	23.43	240.36	279
25-28	573.12	1290.97	1276.60	675.26	482.84	529.59	24.56	366.95	174
29-32	0	0	0	0	0	0	0	0	0
33-36	0	0	0	0	0	0	0	0	0
37-40	0	0	0	0	0	0	0	0	0

. Sample Size - the number of 30 second intervals observed

. All categories except #Users and Sample Size are computed means

TABLE 6d

CP MONITOR DATA FOR APPROACH 2 - THURSDAY

*UNITS IN 100th's of SECONDS

# USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	PGEXCP	PGREAD	VSIOU	VMIOU	SAMPLE SIZE
1-4	0	0	0	0	0	0	0	0	0
5-8	0	0	0	0	0	0	0	0	0
9-12	221.90	424.26	2389.58	69.32	33.07	33.72	3.31	58.86	80
13-16	324.73	720.82	2003.85	187.74	99.20	100.17	11.51	169.14	248
17-20	411.64	1018.17	1635.27	282.90	180.91	189.80	37.36	202.42	408
21-24	489.02	1166.96	1411.34	314.91	226.31	243.02	41.92	240.77	388
25-28	570.17	1332.96	1193.51	443.18	353.27	394.48	82.43	272.72	68
29-32	0	0	0	0	0	0	0	0	0
33-36	0	0	0	0	0	0	0	0	0
37-40	0	0	0	0	0	0	0	0	0

. Sample size - the number of 30 second intervals observed

. All categories except #Users and Sample Size are computed means

TABLE 6e

CP MONITOR DATA FOR APPROACH 2 - FRIDAY

*UNITS IN 100th's of
SECONDS

# USERS	*CPTIME	*PROBT	*WAITT	*WTPAGE	PGEXCP	PGREAD	VSIOU	VMIOU	SAMPLE SIZE
1-4	0	0	0	0	0	0	0	0	0
5-8	0	0	0	0	0	0	0	0	0
9-12	373.33	173.50	2489.25	164.25	57.33	62.42	155.66	165.08	12
13-16	279.97	494.95	2267.40	238.28	113.21	122.33	10.90	113.36	42
17-20	400.35	783.77	1869.22	341.48	177.13	187.84	95.87	174.90	31
21-24	522.37	1337.46	1245.55	529.51	403.67	441.20	32.30	370.43	266
25-28	668.67	1465.08	1149.22	998.20	706.66	769.58	28.35	298.64	378
29-32	770.29	1523.72	1016.76	1006.29	767.19	842.64	27.65	380.41	51
33-36	0	0	0	0	0	0	0	0	0
37-40	0	0	0	0	0	0	0	0	0

. Sample size - the number of 30 second intervals observed

. All categories except #Users and Sample Size are computed means

Approach 1

In Fig. 17, the CPU wait time and page wait times were plotted against the number of users. The page wait time is seen to rise as the number of users increases until it consumes almost all of the wait time at about the 28-30 user bracket. This suggests a bottleneck in the paging subsystem.

Looking at the system again, main storage has 512K bytes with the primary paging device (a 2301 drum) supplying 4 M bytes. Secondary paging is supplied by two CP system areas designated on two Potter 2314 disk units with a total of 138 cylinders or approximately 16 M bytes. This storage area is also utilized for CP spooling operations. No information was available as to how the system divides this area. Addressable virtual storage in the system is 2^{24} locations or roughly 16 M bytes. Of the 512K bytes in main memory, CP nucleus takes 80K bytes, CP work area takes 48K bytes, and CMS save area takes another 72K bytes, leaving 312K bytes of usable main memory for pages.

If this 312K bytes of real memory is added to the 4M bytes of primary drum storage and divided by the default size of 256K bytes of each virtual machine (which assumes a minimal situation) 16.84 users could be accommodated before drum overflow would take place. Of course, this number of users would decrease as the number of users with virtual machine sizes greater than 256K bytes increased.

Therefore, it is not surprising that paging becomes a problem as the number of users increases and more time is spent going out to the secondary paging device.

FIGURE 17 - PAGING SUBSYSTEM

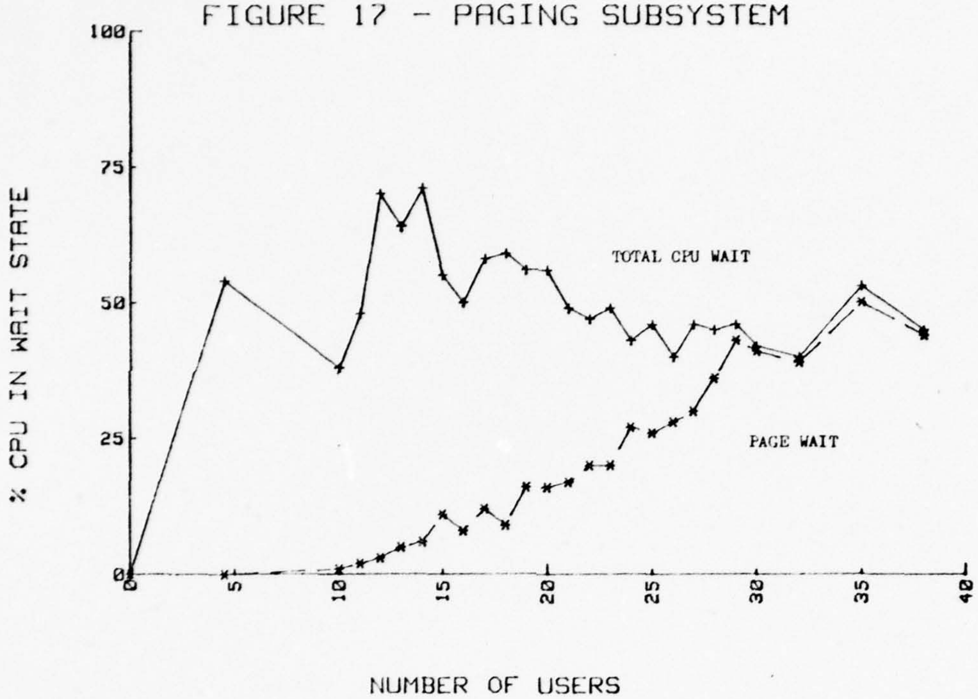
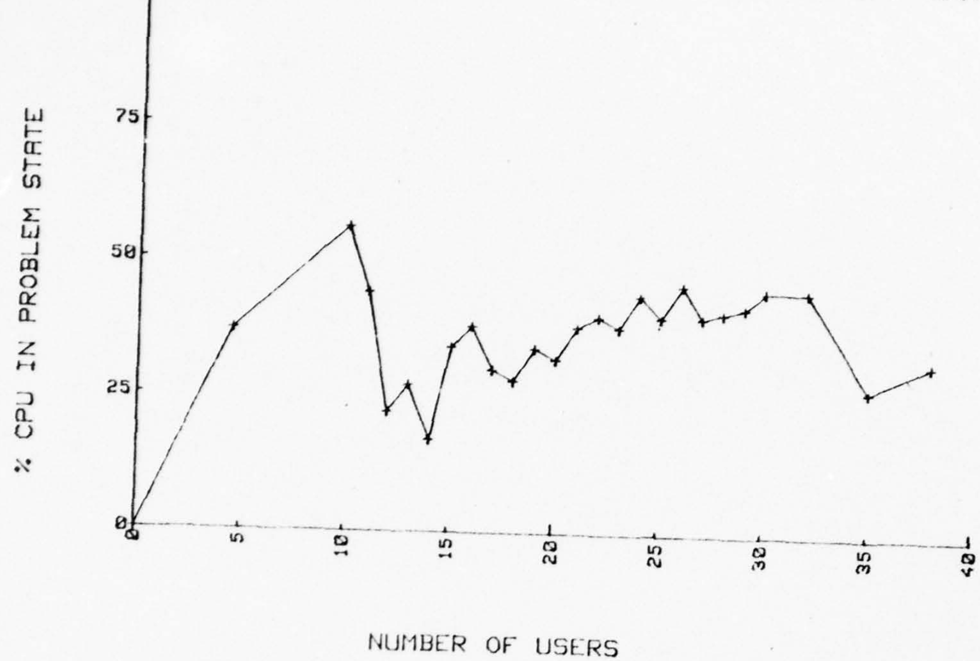


FIGURE 18-CPU PROBLEM STATE



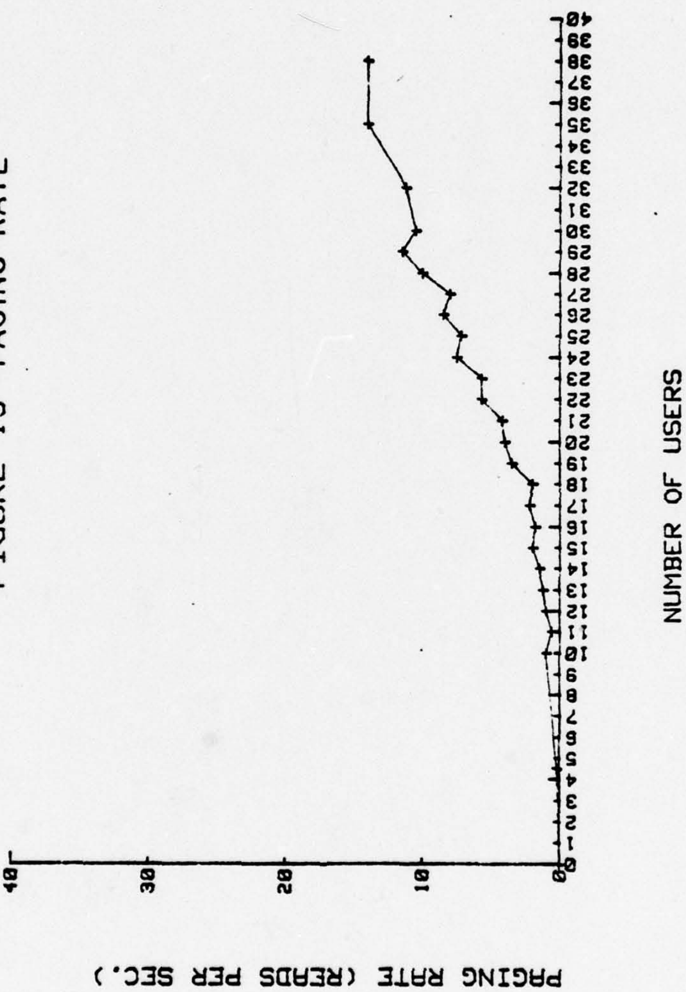
AD-A070 223 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
COMPUTER NETWORKS.(U)
MAR 79 R B KURTH

UNCLASSIFIED NL

2 OF 2
AD
A070223

END
DATE
FILMED
8-79
DDC

FIGURE 19 - PAGING RATE



In Figures 18 and 19, where CPU problem state time and paging rate (pages read/elapsed time in seconds) is plotted against number of users. The paging rate increases as the number of users increases while the CPU problem state time, although fluctuating somewhat, tends to also increase. This at least suggests that the CPU is not saturated with overhead due to paging.

Approach 2

For this approach the data was analyzed on a daily basis, attempting to give a more stable picture. The assumption was made that workload representations might be less variable if looked at on a daily basis.

In order to obtain enough observations (minimum of 50) for each grouping, the partitioning had to be adjusted from the previous approach. Figures 20a, b, c, d, e show total CPU wait time and page wait time plotted against these groups for each of the 5 days. Not all user groups are represented for each day. As was seen in the previous case, the page wait time increases until it consumes a significant portion of the CPU wait time, indicating a paging subsystem bottleneck due to primary paging device saturation. Also on all five days the CPU problem state time increases with the number of users (Fig. 21) suggesting, as before, that the CPU is not saturated due to paging.

Summary

Although only a small subset of the analysis procedure was possible due to system measurement limitations, an insight

THE USER GROUPINGS FOR FIGURES 20a, b, c, d, e are broken down as follows:

NUMBER OF USERS	USER GROUPINGS
1-4	1
5-8	2
9-12	3
13-16	4
17-20	5
21-24	6
25-28	7
29-32	8
33-36	9
37-40	10

FIGURE 20A - MONDAY

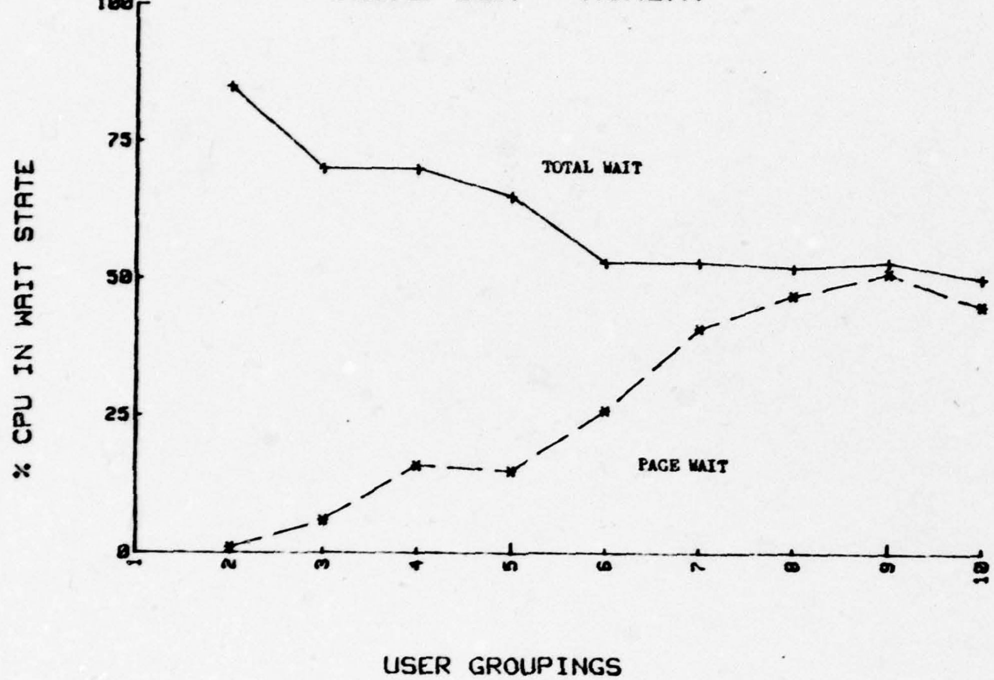


FIGURE 20B - TUESDAY

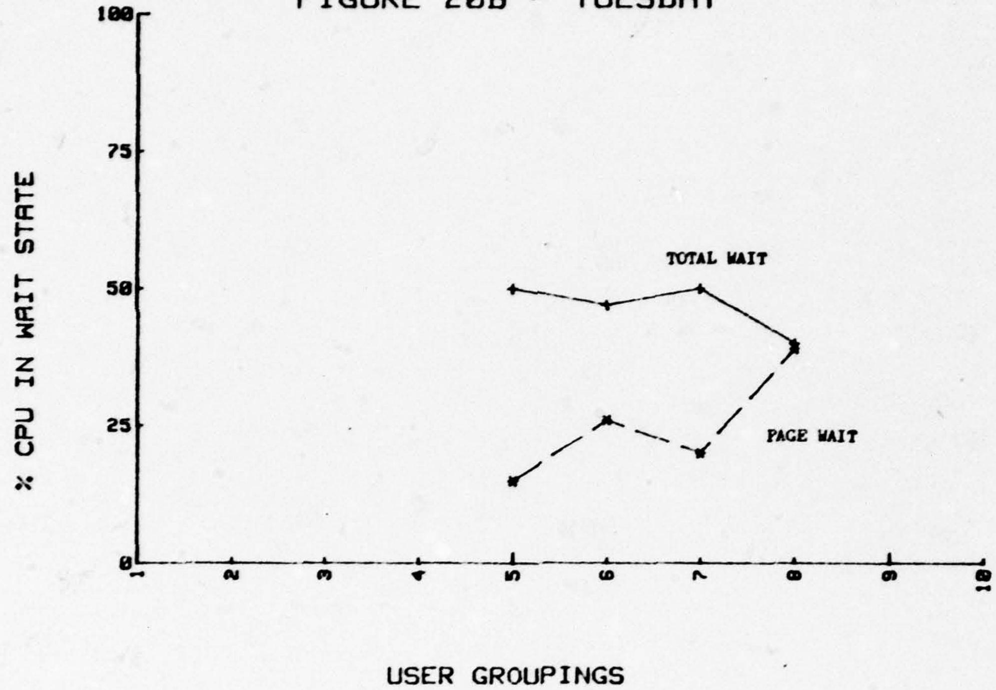


FIGURE 20C - WEDNESDAY

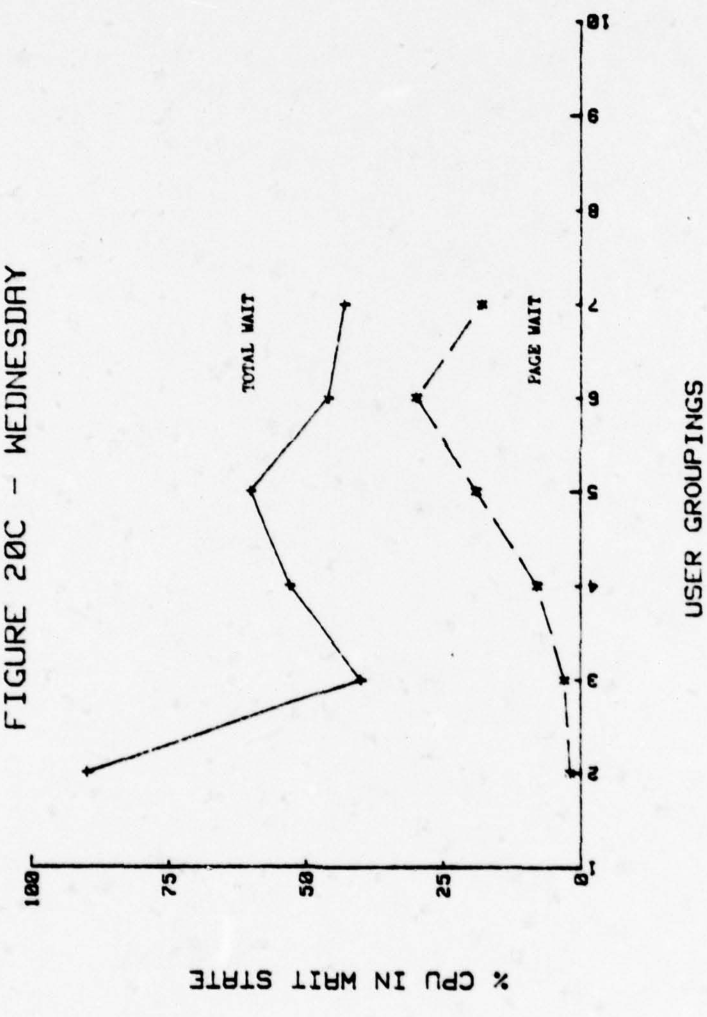
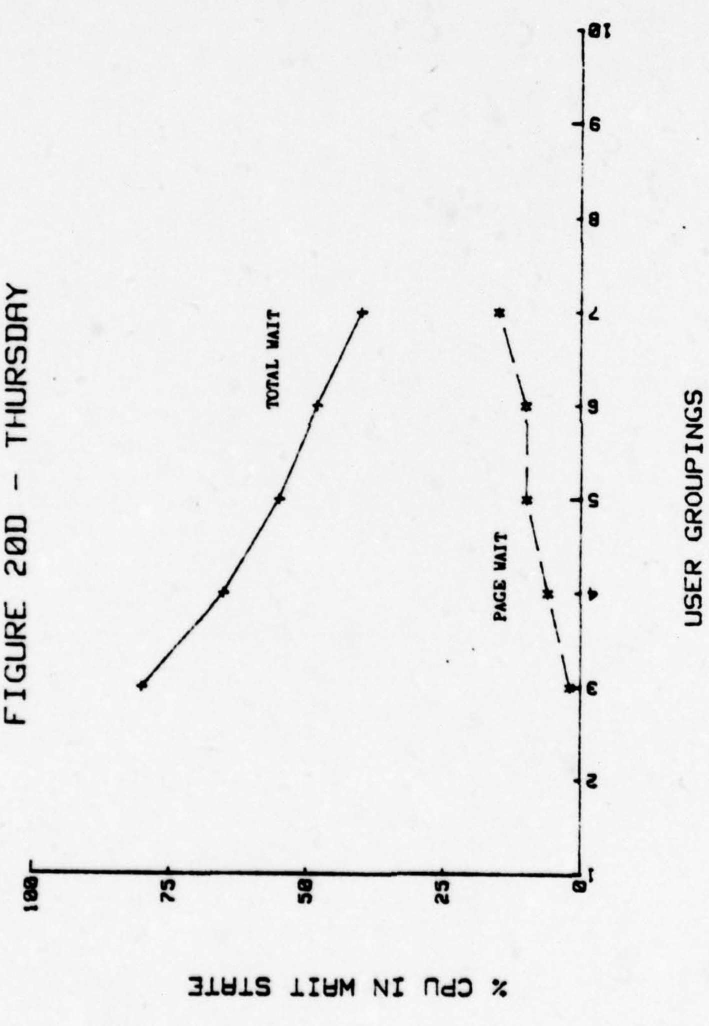


FIGURE 20D - THURSDAY



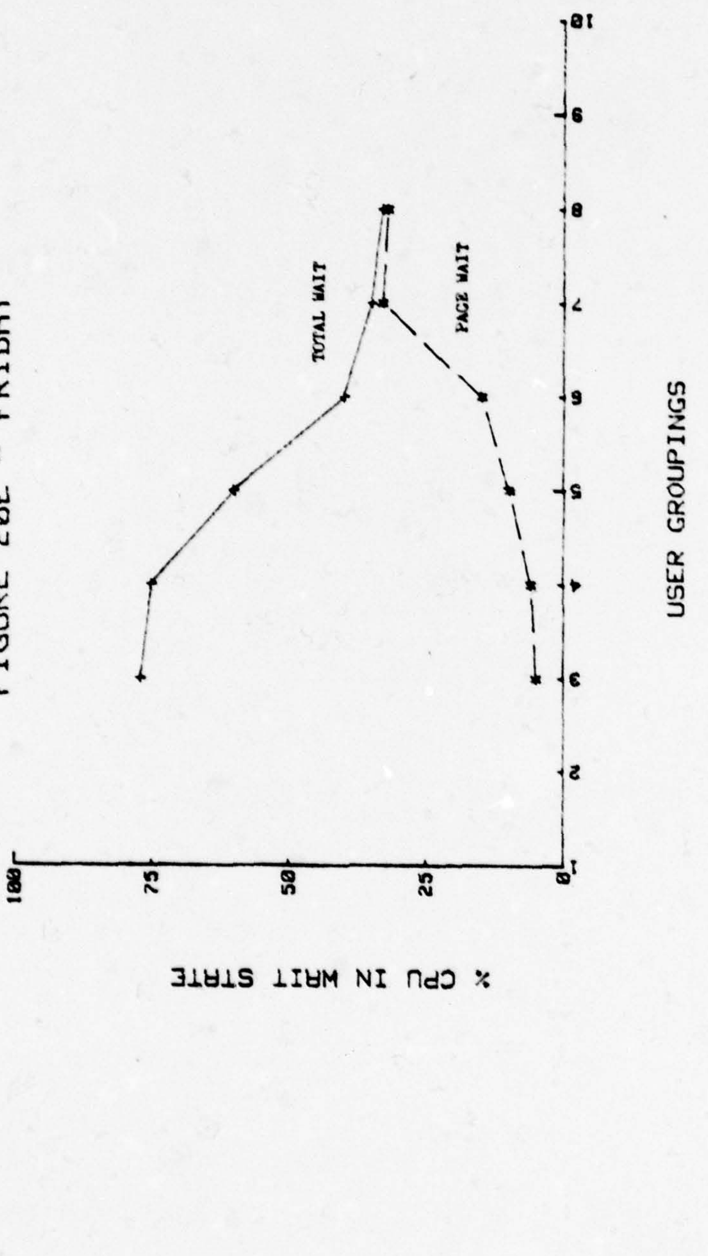
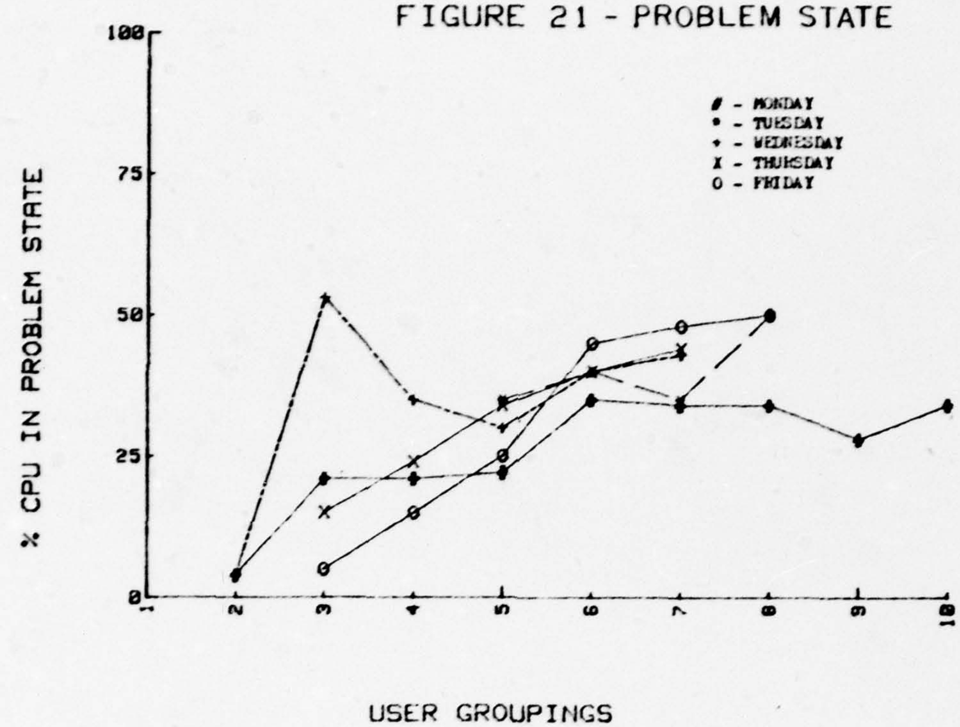


FIGURE 20E - FRIDAY

FIGURE 21 - PROBLEM STATE



was gained into system performance. Two useful pieces of information were found.

- the system has a paging subsystem bottleneck, and it seems reasonable to conclude that the primary paging device (IBM 2301 drum) is at the root of the problem. The system should have enough primary paging storage to handle the virtual memory size in the system. Addition of at least one other drum would certainly do much to alleviate the excessive system response time users are experiencing.
- Data pertaining to user levels over specified timing periods was gathered which represented typical days of operation. This data will be used later in order to predict requirements for future systems.

IV. NETWORK DESIGNS FOR NPS ENVIRONMENT

A. INTRODUCTION

In the preceding chapters, the diverse computing environment at NPS has been discussed. These different user needs have not successfully been met by the one large central mainframe concept. It seems reasonable that purchasing another newer but larger mainframe to process both the research and educational requirements would have similar problems in the future.

The network approach on the other hand has worked very well in similar campus environments. (Ref. 8) It provides the type of system modularity that can keep pace with new hardware technology and it can offer specialized computing for research projects without sacrificing the generality of the system. Therefore, it seems that a network might be a viable alternative for the NPS campus. In the following sections two networks will be presented in configurations with the NPS campus specifically in mind.

B. DESIGN OBJECTIVES

Before any designs may be presented the motivational and functional goals derived from knowledge gained through previous chapters and specific insights into the desires of user and operational groups must be established.

This forces the organization to set down and formulate reasonable objectives for their new system based on their

current and projected needs. From this information, the designer, ideally, will be able to effect a better design to meet the organizational objectives. Whether or not he succeeds, the organization will at least have a yardstick by which to compare proposed designs.

Some of the motivational and functional goals for the NPS system are listed below.

MOTIVATIONAL GOALS

a. Capability

- provide adequate computing and storage capacity for batch and time-sharing users with emphasis on research requirements
- provide acceptable turn around time for batch jobs
- provide acceptable response time for time-sharing
- provide for high system availability
- provide for high level of resource sharing with utilization of existing hardware and software
- provide for integration of batch and time-sharing
- provide support for a high number of graphics devices
- system should be capable of being used in its own development.

b. Evolvability

- system should be adaptable to changing military and research needs; this infers modularity
- continuity of user interfaces should be maintainable even though system changes occur

- provide for hardware upgrade path for at least the next eight years

c. Convenience of use

- system should be easy to learn and use
- accessibility should be widespread with no fundamental incompatibilities between interactive and non-interactive use
- understandable debugging tools should be available
- tutorial packages for system use and as programming aids should be available
- help function mode for time-sharing users as well as escape mechanisms should be available
- a systems capabilities listing with abstracts for software library routines should be available

d. Reliability

- MTBF and MTTR figures should meet availability requirements for the system
- graceful degradation of system functions should be designed into the system for hardware and software

e. Flexibility

- entire system should be highly reconfigurable; the system should not come to a halt in case of scheduled or unscheduled maintenance
- easy connection of micro and mini processor based systems in NPS laboratories should be provided for

f. Efficiency

- system should be as efficient in time-sharing mode as in batch and vice versa so that research approaches will be independent of cost considerations
- the system should make use of all its capabilities of its hardware components

FUNCTIONAL GOALS

a. Information Processing

1. Hardware functions

   - batch processor should be 10 times faster than current CPU
   - batch storage capacity should be 4 times greater than current system
   - a minimum of 1 B bytes of direct access storage exclusive of operating system and work file requirements for user files is needed
   - total system storage will be a minimum of 100 M bytes of logical address space
   - terminals
     . existing terminals will remain in the system
     . additional synchronous terminals operating in page mode at up to 9600 bps are needed
     . provide initial installation of 100 terminals (existing and new) and additional 100 during 1980-1990 time frame.
     . some terminals should have graphics capabilities with hard copy peripherals

  - other capabilities of terminals should include editing in page mode, scrolling, plug-in character sets, magnetic cartridges, and highlight of fields
- terminal functions
  - mail system for leaving messages should be available
  - scanning and modification by both context and line number with local and global modification mechanisms
  - option for defining both horizontal and vertical limits for global scan and modification
  - all user input should be monitored in order for editor to immediately flag format errors, whether the input is new data or changes to existing data
- symbolic debugging
  - interactive debugging; capability of suspending and restarting or cancelling program execution with mechanism to selectively display status and contents of main storage; corrections could then be made and program restarted.
  - trace statements as they are executed and trace variable changes
- desk calculator mode
- commands to invoke canned application packages

such as statistics, mathematics and engineering packages both tutorial and interactive computation

- support for graphics and plotting
- multiple function capability; e.g., a user can be compiling a program while doing some other function
- character set conversion routines
- task management
  . interprocess communication
  . intraprocess communication
  . modifiable scheduler to enable tuning of throughput parameters
- resource management
  . flexible hardware component assignments (if a component is down system automatically looks for another eligible, available component)
  . hardware utilization should be balanced across sytem components, as much as possible
- utility
  . input/output spooling
  . data base management system with relational and hierarchial schemes
  . dynamic linking of modules
  . dynamic memory management
  . cross calls to languages
  . system accounting package

x multilevel tree structure

x password protected

x dynamic updating of accounting data

x accounting data should be logged to a file to provide good audit trails

x provide post-processing program to handle accounting data

b. <u>Network Processing</u>

1. Hardware Function

- concentration
  - . no defined requirement
- coupling
  - . modems will continue to be used for existing terminals - rate capabilities follow
    - x asynchronous up to 1200 bps
    - x synchronous up to 9600 bps
- distribution
  - . all connections will be point to point
  - . processor to processor communication will be at a minimum rate of 50 K bps
- switching
  - . new terminals including RJE's will be controller or computer switched
  - . it will be possible to logically connect any terminal to the processor subsystem for the execution of a given job, independent of the location or physical connection of the terminal

- Source/Destination interface
  - ARPANET interface will be provided
  - 3 Defense Manpower Data Center RJE sites will continue to be supported

2. Software Functions

- Routing
  - interactive terminals will be able to route jobs by means of system commands through the network (e.g., user sends program file built during terminal session to the batch computer for execution)
  - other attributes of the routing function as specified in Chapter I.

Other functions-integrity, journeling, statistics, utility, and supervisor control - are generally required to support other functions, as specified in Chapter I.

A number of these software functions will depend upon the hardware available and monetary constraints. For NPS applications these functions will be presented in terms of the architectural configuration presented, e.g., types of software functions which will be required to support a centralized data scheme.

d. Proposed designs

In this section two designs will be considered. Each will be discussed generally in terms of some strengths and weaknesses based on the motivational and functional goals specified in the preceding section. At the end of this dis-

cussion each design will be evaluated based on a more generalized subset of design objectives for NPS. These objectives are modularity, evolvability, reliability, graceful degradation, and ease of operation. For the purpose of this analysis these terms are defined as follows:

modularity - the ability of a system to easily absorb growth by allowing expansion to occur smoothly

evolvability - the system should be adaptive to changes in the nature of research and military needs, as well as technological changes

reliability - the probability that no failure will occur that will put one or more sites out of operation

graceful degradation - ability of the system to gracefully absorb system failures and continue to operate at some reduced performance level

east of operation - view of system operation from the operators' and users' viewpoint, i.e., the system should provide simplicity of use with minimal operator intervention

Each of the designs will be geographically contained within the confines of the rectangle formed by Ingersoll, Root, Spanagel, Bullard, and Halligan Halls as depicted earlier in Figure 7. The external dimensions of this rectangle are 317 meters (Ingersoll-Spanagel) by 97 meters (Root-Halligan).

Each design will primarily concern itself with the time-sharing subsystem and the batch-time sharing integration

problem. The existence of the maxi batch computer will be assumed and will only be specifically discussed if it impacts the time-sharing subsystem.

C. DESIGN 1

Figure 22 shows a simplified diagram of design 1. The major components in this configuration are the maxi batch computer supported by a mini subnet for time-sharing.

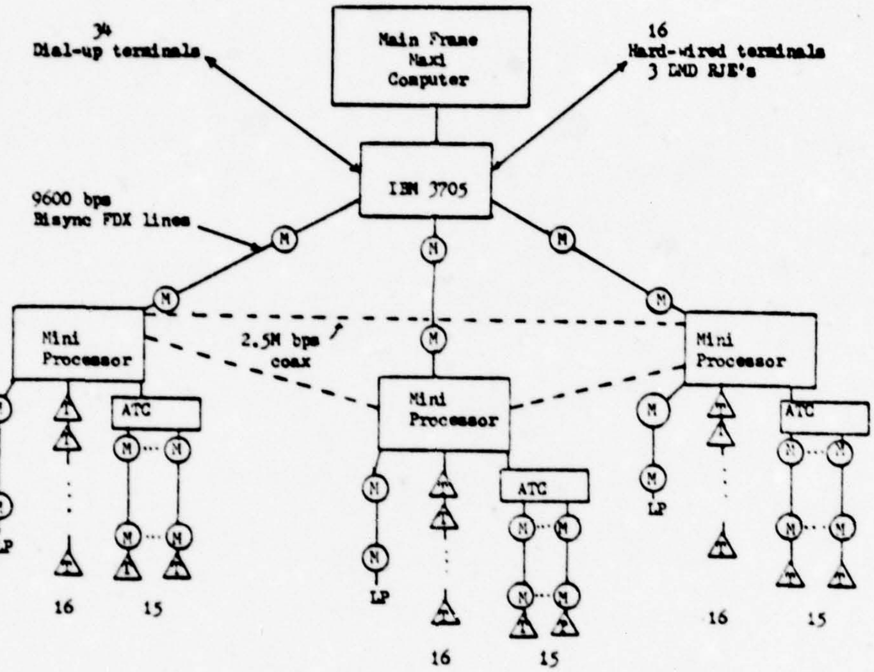
The maxi batch computer will continue to support existing terminals through the IBM 3705 since all connections already exist. This will consist of approximately 50 terminals.

One immediate problem surfaces here in that the maxi time-sharing system and the subset time-sharing systems will require users to know two command languages and to be knowledgeable about subtle language differences. This is not viewed as an insurmountable problem.

The subnet may communicate with the maxi batch computer on a bisynchronous full duplex line at 9600 bps. Each mini emulates an RJE site. This allows users to build files using the subnet and pass them to the maxi batch computer for processing and subsequent output. Jobs may be queued to insure effective utilization of the batch/time-sharing integration links.

In the subnet each mini will be self-sufficient with a stand-alone capability. Each will have its own operating system and storage for local files.

This can include up to 2 M bytes of main memory and an additional 960 M bytes of direct access storage at each mini. Magnetic tape units with 800 and 1600 byte/inch density are also available.



ATC—asynchronous terminal controller
LP - lineprinter
M - modem
T - terminal

FIGURE 22 - Design 1

In addition computer to computer communication including interprocess and block transfer communication are possible on a coax line at rates up to 2.5 M bps. All three minis are linked in this fashion.

There will be three RJE sites in the subnet; Spanagel Hall, Root Hall, and Halligan Hall. Each site will have a medium grade line printer operating at 600 lpm.

The minis themselves will be located in Ingersoll Hall for reasons of ease of maintenance, operational considerations, and security.

Terminal clusters in addition to the ones already provided will be located in Ingersoll Hall, Spanagel Hall, Root Hall, Bullard Hall and Halligan Hall. There will be 45 dial up asynchronous terminals operating up to 2400 bps and 48 hardwired synchronous terminals operating up to 9600 bps. Clusters of these terminals will be located as follows:

	Mini 1		Mini 2		Mini 3	
	A	S	A	S	A	S
Spanagel	4	5	4	5	4	6
Root	3	2	3	3	3	3
Halligan	3	3	2	3	3	2
Bullard	2	3	2	3	3	2
Ingersoll	3	3	3	2	3	3

The asynchronous terminals are connected in a point to point configuration while the synchronous terminals are connected in a multipoint daisey chain with a power down bypass cable for bypassing a terminal which has failed.

There is nothing which will prevent the existing dial up terminals from accessing the subnet.

The way in which the system operates enables any terminal user regardless of his physical connection to the subnet access to any subnet file/data base. He will be able to operate on this file as if it were located locally without transfering it to his remote location.

The subnet also provides the same type of flexibility for device selection. A user may access any device in the subnet from his remote location. Both file and device remote access take place through the computer to computer links.

The implementation that enables the subnet to function properly is based on functional layers of software with only the high level system services visible to the user. The layed concept is very similar to IBM's System Network Architecture (Ref. 24). Each internal layer provides flexibility for additions without requiring alternation of application programs. The layers of software would look similar to the list below with the deepest level last.

User Language Programs

- application programs in high level languages for both users and system development

Network Access Method

- gives users access to system files, devices, and data bases and processes network commands

Network Manager

- is responsible for managing network functions such as network error recovery, and network topology.

Message Control Protocol

- this layer provides control functions addressing information, message type, and other requirements to assure correct message transmission

Communication Line Protocol

- provides grammer by which two or more rules systems may communicate

Communication Line Controller

- this consists of the hardware which connects lines and the software driver for the hardware interfaces.

Analyses

Modularity - Modularity of the system is good. New modules may be added to the system without the need for duplicating devices at each mini since remote file and device access is provided for. The number of terminals in the subnet with this configuration may be expanded to 276, but this is a maximum and may degrade system performance.

Evolvability - Evolvability of the system is good. Each mini unit can be specially tailored for a specific purpose if the need arises yet users attached to it may still access the more general side of the subnet and vice versa.

Reliability - Reliability of the system is fair to good. It provides users time-sharing services if the maxi batch computer fails. These services begin to decay as minis begin to fail since devices and files at those locations will not be accessible. Although users may go to another site to obtain access to the system they may not be able to access all their

files. This is where cassette tape capability of terminals in the system would be nice. A user could then take his file with him to another terminal.

<u>Graceful degradation</u> - The system provides fair graceful degradation of facilities. When one mini fails all the files and devices associated with it cannot be used. On the other hand, only a portion of the system's users are affected.

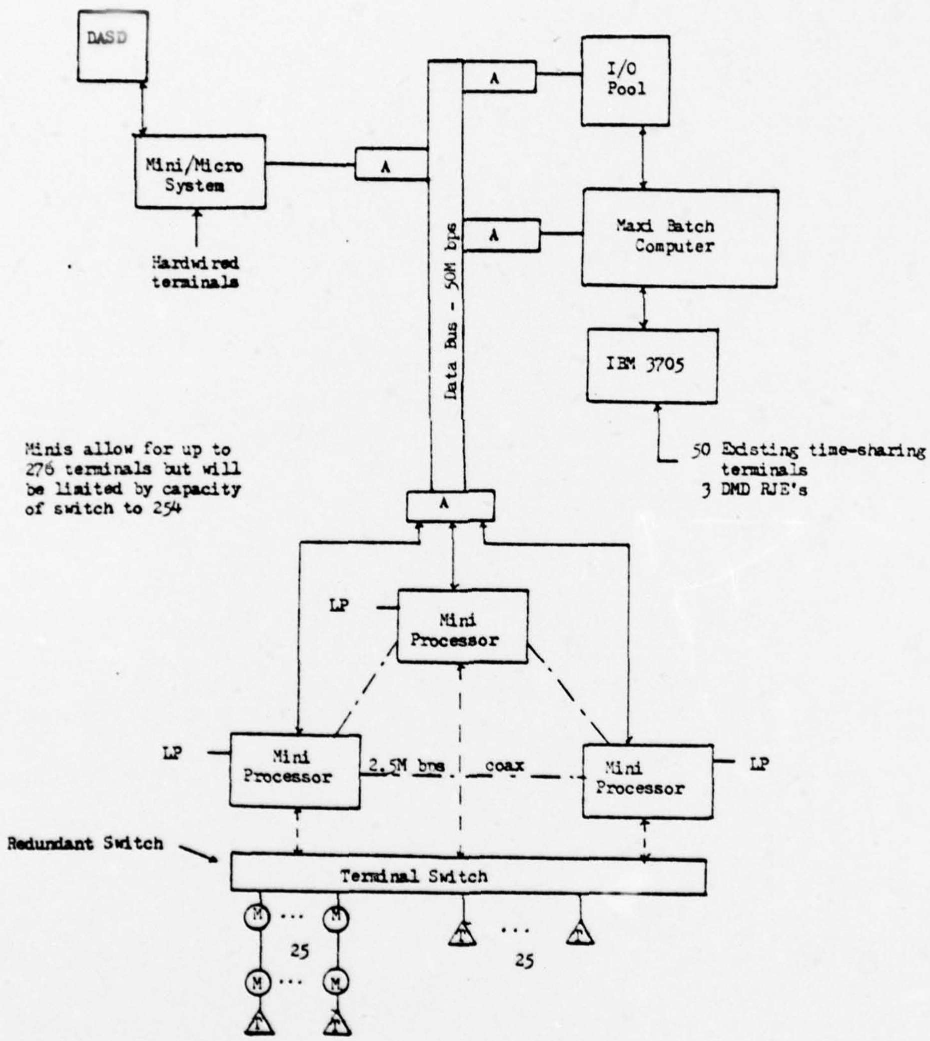
<u>Ease of use</u> - Since all major hardware components will be centrally located, operator personnel will be minimized and the ease of use is good. From the users standpoint the only drawback is the existence of two time-sharing systems. However, accessibility is so great that the user should be able to do all of his work on a single system, if he desires. The user should be able to do all functions previously done with cards faster and with the advantage of an editing capability.

D. DESIGN 2

Figure 23 shows a simplified diagram of design 2. The major difference in this approach to design 1 is the data bus.

Most design difficulties occur in networks when interfacing different vendors equipment so that the network can move information from one device to another. One of the major problem areas is what to do when a new hardware component is added to the system. Do you have to change all the interfaces to integrate it into the network? Hopefully, the answer is no. This is where the data bus concept excels. It allows the designer to develop the network around the data bus which is composed of high speed digital transceivers utilizing a coaxial cable.

FIGURE 23 - Design 2



The adapter unit is what makes the data bus approach possible. It is divided into four logical sections; the data bus interfaces, a microprocessor, buffers and control logic, and the equipment interface.

Each adapter can interface up to four data bus coax cables. Each coax cable allows up to a 50 M bps (36 M bps effective) data flow.

The microprocessor handles equipment functions and responses and manages data flow between the equipment and the adapter buffers.

The control section contains the logic to resolve contention for use of the buffers by the data bus and equipment interfaces. The 100 M bps buffers allow a 50 M bps data movement simultaneously at both the data bus and equipment interfaces. When the buffer is half full, it transfers the data on the bus in a burst mode. There are registers in the control section that contain buffer address and length of data transmission. Other registers provide network addressing information. The buffers allow each adapter to match data rates of the attached equipment.

The equipment interface provides electrical and cable interface to the specific attached equipment. In some cases it also provides assembly/disassembly, holding registers and hardware ready-resume logic.

A network message format is used to transmit all data in the network. The data flow takes place in three asynchronous steps:

- sending equipment to adapter buffer
- adapter buffer to receiving buffer
- receiving buffer to receiving equipment.

All three steps can take place simultaneously, for different data transmissions. With long data streams an alternating buffer scheme is utilized to insure continuous data flow. The data bus itself is only used when the data is burst on line so that it can be shared among other adapters. The adapters implement the communication protocol needed to control traffic flow by adapter logic and microprocessor routines, totally decoupled from the attached equipment.

Some of the benefits of this data bus approach are multiple equipment interconnection, logical and physical distributions of functions, improved input/output efficiency, sharing of resource and data files, dynamic device reconfigurations, and high speed data transmission.

In the NPS system, the design is organized around the data bus. All files and peripheral units will be centralized in shared I/O and mass storage pool. The mass storage pool will be protected from failure by a redundant controller. The maxi batch computer, mini timesharing subnet, and the laboratory mini/micro systems are all interfaced to the data bus by appropriate adapters. The maxi batch computer as in design 1 will support existing terminals in their present location. Thus, the two time-sharing system problem exists in design 2 also.

The mini subnet will be interconnected in the same manner as design 1 with the following exceptions

- the terminals will be connected to the mini subnet via a terminal switch
- the terminal switch should be a redundant, if not, complete failure of this switch will drop subnet terminals off line
- minis will not maintain local user files - they will reside in mass storage
- if desired, the RJE capabilities of the subnet may be connected via the IBM 3705 to provide redundancy
- the subnet terminal capacity will be limited from 276 to 254 by the terminal switch.

The terminal clusters will be located in the same manner as in design 1. Only 50 terminals were included in Fig. 23 for simplicity, but this is expandable up to 254 asynchronous and synchronous terminals.

The design, except for the added data bus features which enhance the network, will look the same to the user as design 1.

<u>Analysis</u>

<u>Modularity</u> - the modularity of the system is excellent. It only requires the existence or development of the appropriate adapter to integrate any new module to the system without considerations of how they impact other units.

<u>Evolvability</u> - the evolvability of the system is excellent. Specialized equipment may be added or deleted from the data bus as requirements change without affecting other modules.

<u>Reliability</u> - The reliability of the system is good. Two modes of time sharing are offered through two different

switches. It is recommended that the mini subnet terminal switch be a redundant switch, since the major part of the time-sharing system (150 terminals), when expanded, will depend on this switch.

Graceful degradation - The assumption will be made that the terminal switch is redundant in this discussion. Assuming this, the system provides good graceful degradation. The following list of failure-outcomes is provided to summarize how the system reacts.

Failure of	outcome
any mini -	RJE function for that mini is lost; users are switched to remaining minis
terminal switch -	redundant switching mechanism is used
RJE terminal -	user must go to another RJE site
mini adapter -	subnet may operate stand-alone but user files not resident will be inaccessible unless RJE function of subnet is connected to IBM 3705.
maxi batch computer-	batch processing and time-sharing oriented with batch computer is lost

This list could continue but it is obvious that unless several components fail at the same time the system is not seriously disrupted.

Ease of use - Ease of use as in design 1 is good. All hardware except terminals, modems, and RJE line printers will be located at the main site (Ingersoll Hall).

Users will enjoy the same benefits derived from design 1, except that devices and files will be pooled in a centralized location rather than divided among the minis. The pooling will enable users on a mini which fails, to be switched to another mini without possible loss of their files.

This system may be more powerful than what is truly needed at NPS. The data bus, adapters and associated software are not cheap, but if such a system is affordable, it certainly gives the best performance opportunities for changing research requirements.

F. CONCLUSIONS

In the first chapter of this work network structures, building blocks, and design criteria were discussed on the assumption that a network approach would be best suited for NPS computing environment.

In the second chapter, performance measuring tools and the man-machine interface impact on performance in time-sharing systems were discussed. From the knowledge of these elements, a methodology for analysis of the current NPS system was extracted.

The third chapter gave background information for readers not familiar with the NPS system and then profiled the user groups of the time-sharing system. The last portion of the chapter presented an analysis of the time-sharing system using one type of performance measuring tool specified in the previous chapter. From the information presented in this chapter

and other external inputs, the design goals of the network were specified in the fourth chapter.

These design goals were outlined in terms of the network design methodology presented in chapter one. The methodology could not be carried out to its logical end since specification of hardware and software for the network was beyond the scope of this work.

Nevertheless, two network designs were presented based generally on the design objectives. Both designs were similar with the second utilizing a more risky architecture.

The first design is somewhat more conservative but does not have the flexibility and future growth potential.

Both designs were based on available hardware and software; both are feasible. If you are willing to take risk and pay a little more with the promise of greater growth potential in future years design 2 should be selected. If not, design 1 should be your choice.

W. R. CHURCH COMPUTER CENTER
NAVAL POSTGRADUATE SCHOOL

APPENDIX A

MAJOR SOFTWARE USED ON IBM 360/67 AT NPS

I. Programming Languages

OS/360 MVT Rel 21.8B
HASPII

ASMF

ASMG

FORTRAN IV G

FORTRAN IV H

FORTRAN IV H Extended

WATFIV

COBOL ANSI

COBOL 4

WATBOL

PL/1, Version 5

PL/1 OPTIMIZER

PL/C

BASIC

ALGOLE

ALGOLW

PASCAL

PL/M8&80

LISP 1.5

XPL

XPL4

PL/360

SNOBOL4

SPITBOL

ASSIST

CP-67/CMS Vers. 3.2

ASMF

FORTRAN IV G

COBOL 4

PL/1, Version 5

BASIC

ALGOLE

ALGOLW

APL/360 (VS APL later)

XPL

PASCAL

PLM/M8 (Intel 8008)

PL/M80 (Intel 8080)

PL/360

SCRIPT

APPENDIX B

Additional Software Resources (Source indicated in parentheses)

SIMULATION			
	Continuous	CSMP-III	Continuous Simulation Modelling Package (IBM)
		DSL	Digital Simulation Language (IBM)
		DYNAMO II	(MIT)
	Discrete	CPSS-V	General Purpose Simulation System (IBM)
		SIMSCRIPT II.5	(CACI)
STATISTICS			
		BMD	Bio-Medical Package (UCLA)
		BMDP	
		SPSS	Statistical Package for Social Sciences (U. of Chicago)
		SAS	Statistical Analysis System (North Carolina State University)
		STATPACK (APL)	Statistical Package (U. of Alberta)
		SNAP/IEDA	Interactive Exploratory Data Analysis (Princeton)
		OSIRIS	U. of Michigan Survey Research Center
SYMBOLIC MANIPULATION			
		REDUCE2	U. of Utah
		FORMAC(PL/1)	Formula Manipulation Compiler (IBM)
		FORMAC(FORTRAN)	" " " (IBM)
LIBRARIES			
		SSP3	Scientific Subroutine Package (IBM) (supplemented by locally written routines)
		IMSL	International Mathematical & Scientific Library (IMSL)
		Programming Aids Library (SHARE, etc.)	OS Utility Routines
		EISPACK	Eigensystem Package (Argonne Lab.)
		FUNPACK	Function Package (Argonne Lab.)
		LINSYS	Linear Systems (U. of Victoria, B.C.)
GRAPHICS			
		CALPLOT	Drum Plotter (Cal Comp 765)
		CSP	Graphical Subroutine Package, IBM 2250
		PLOT-10	Tektronix Graphics(Tek 4012 under CP/CMS)
		SYMAP	Computer Mapping (Harvard)
		PLT301	3-Dimensional Isometric Plotting
ENGINEERING			
		SAP	Structural Analysis Package(UCLA,UCB)
		NONSAP	Non-Linear " " " "
		ECAP	Electronic Circuit Analysis Package(IBM)
		LISA	Linear Systems Analysis (NPS)
		NASAP	Non-Linear Systems Analysis Package(NPS)
		ROOTLO	Root Locus (NPS)

APPENDIX C

ACCOUNTING & MEASUREMENT

SMF	System Management Facility (IBM)
SARA	System Analysis and Resource Accounting Program (Boeing)
PROGLOOK	Monitor Execution of a Load Module (SLAC)
PROFEELII	Monitor Execution of a Load Module (CACI)
SLACMON	OS System Monitor (SLAC, Stanford)
MEASURE	CP/CMS System Measurement(SUNY, Stony Brook)

OTHER

SDC	Systems Design Game (Cornell)
MPS/360	Mathematical Programming System (IBM)
ICAP	Integrated Carrier ASW Prediction System
SORT/MERGE	(IBM)
TPS	Text Processing System (OS/360)
NSCRIPT	Manuscript Preparation (CP/CMS)
FAST	Force Structure Simulation Model
WEIS	World Events Information System (USC)
	Multiple Precision Arithmetic Package (Stanford)
MARK IV	Report Generator

LIST OF REFERENCES

1. Computer Center User's Manual.

2. Bard, V., "Performance Analysis of Vertical Memory Time-Sharing System," IBM System Journal 1975, No. 1.

3. Herndon, Edwin S., "Computer Systems Planning - Tools & Techniques," 1976 National ACM Conference Proceedings, p. 203-207.

4. Cameron, James S., "An Approach to Benchmarking Terminal Oriented Systems," 1976 National ACM Conference Proceedings, p. 208-212.

5. Miller, Lawrence H., "A study on man-machine interaction," AFIPS 77 Vol. 46, p. 409-421.

6. Boies, S.J., "User behavior on an interactive computer system," IBM System Journal No. 1 1974, p. 2-18.

7. Mamrak, S.A. and P.A. DeRuyter, "Statistical Methods for Comparing Computer Services," Computer Nov 77, p. 32-39.

8. Kuo, Abramson, Computer Applications in Electrical Engineering, 1973.

9. Rollins, Turner and Hank Levy, "Performance Evaluation of IAS on the PDP-11/70," ACM/Sigmetrics 1976, pg. 71-74.

10. "Computer performance measurement and evaluation tools," SHARE 71-73, p. 210.

11. Michlin, J., "A better way to start a computer performance program," SHARE Vol. 1 1977, p. 30-33.

12. Lockett, Jo Ann and Thomas E. Bell, "The Rand Monitor/Stimulus-generator: a simple approach to response time analysis in on-line computer systems," SHARE Vol. 1 1975, p. 75-89.

13. Merrill, B., "Computer performance measurement," SHARE Vol. 1 75 p. 93-97.

14. Watkins, S.W. and M.D. Abrams, "Remote terminal emulation in the procurement of teleprocessing systems," AFIPS 1977, pg. 723-727.

15. "Control Program-67/Cambridge Monitor System," IBM Publication GY 20-0590-1.

16. Chou, W., "Computer Communications Networks - The Parts Make up the Whole," Proceedings AFIPS National Computer Conference, May 1975.

17. Davies, D.W. and D.L. Barber, "Communication Networks for Computers, 1975.

18. Green, W. and U. Pooch, "A Review of Classification Schemes for Computer Communication Networks," Computer Nov 1977 p. 12-19.

19. Beck, H.B., "A Structured Approach to Information Networks," Seventh Annual IEEE Computer Society COMP CON 73, p. 35-38.

20. Loomis, B., "Systems Analyses and Design for General Purpose Computing at the Naval Postgraduate School," CS4200 paper Sept 1977.

21. Schneidewind, N.F., "Analysis of Requirements of Principal NPS batch users," Internal Memorandum to FCPC of NPS, 28 Nov 77.

22. Schneidewind, N.F., "Time-sharing Analysis," Internal Memorandum to FCPC of NPS, 28 Jan 78.

23. Enslow, P.H., "What is a 'Distributed' Data Processing System?", Computer, Jan 78, p. 13-21.

24. McFadyen, J.H., "Systems Network Architecture: An Overview," IBM Systems Journal, No. 1, 1976.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, CA 93940	1
4.	Professor N.F. Schneidewind, Code 54Ss Department of Computer Science Naval Postgraduate School Monterey, CA 93940	1
5.	LCDR Frank Burkhead, Code 52Bg Department of Computer Science Naval Postgraduate School Monterey, CA 93940	1
6.	LT Ronald Brent Kurth, USNR NAS North Island, HC-9 San Diego, CA 92135	1